

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-221996

(43)Date of publication of application : 18.08.1995

(51)Int.Cl.

H04N 1/413

H04N 1/41

H04N 7/24

(21)Application number : 06-010357

(71)Applicant : CANON INC

(22)Date of filing : 01.02.1994

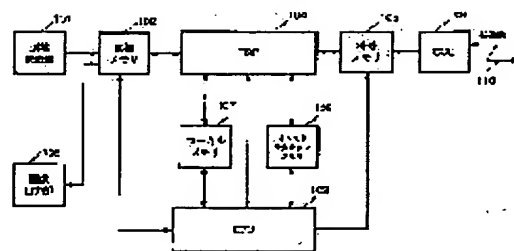
(72)Inventor : NAKAMURA YASUYUKI

(54) METHOD AND DEVICE FOR IMAGE PROCESSING

(57)Abstract:

PURPOSE: To provide the image processing method and its device not requiring the provision of an exclusive processing section for each applied coding system when plural coding systems are applied to the processing method.

CONSTITUTION: A CPU 109 sets a processing program corresponding to an applied coding system to an instruction memory 108 sequentially. A DSP 104 applies coding sequentially to image data stored in a picture memory 102 by the processing program of the different coding system set in the instruction memory 108 and stores the data to a coding memory 105.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-221996

(43) 公開日 平成7年(1995)8月18日

(51) Int.Cl. ⁸	識別記号	序内整理番号	F I	技術表示箇所
H 0 4 N	1/413	D		
	1/41	B		
	7/24			
			H 0 4 N	7/ 13
				Z

審査請求 未請求 請求項の数12 O L (全 35 頁)

(21) 出願番号 特願平6-10357

(22) 出願日 平成6年(1994)2月1日

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 仲村 康幸

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

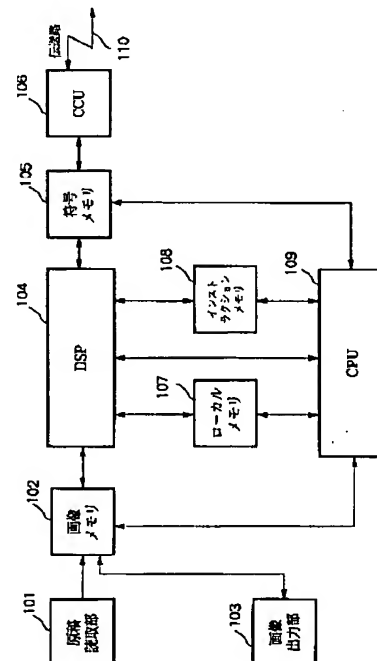
(74) 代理人 弁理士 大塚 康徳 (外1名)

(54) 【発明の名称】 画像処理方法およびその装置

(57) 【要約】

【目的】 複数の符号化方式を適用する場合でも、適用する符号化方式毎に専用の処理部を設ける必要がない画像処理方法およびその装置を提供する。

【構成】 CPU 109は、適用する符号化方式に対応した処理プログラムを、インストラクションメモリ108へ順次設定するDSP 104は、インストラクションメモリ108に設定された異なる符号化方式の処理プログラムによって、画像メモリ102が記憶する画像データを順次符号化し、符号メモリ105へ格納する。



【特許請求の範囲】

【請求項 1】 設定された処理手順に応じて画像データを異なる符号化方式で順次符号化することを特徴とする画像処理方法。

【請求項 2】 前記符号化方式を画像データの所定単位毎に切替えることを特徴とする請求項 1 に記載の画像処理方法。

【請求項 3】 設定された処理手順に応じて、順次、画像データを符号化し、符号データを復号することを特徴とする画像処理方法。

【請求項 4】 符号化と復号とを画像データの所定単位毎に切替えることを特徴とする請求項 3 に記載の画像処理方法。

【請求項 5】 画像データを記憶する記憶手段と、設定された処理手順に応じて画像データを符号化する処理手段と、前記処理手段に異なる符号化方式の処理手順を順次設定する制御手段とを備え、前記処理手段は、前記制御手段によって設定された処理手順に応じて、前記記憶手段に記憶された画像データを異なる符号化方式で順次符号化することを特徴とする画像処理装置。

【請求項 6】 前記制御手段は前記符号化方式を画像データの所定単位毎に切替えることを特徴とする請求項 5 に記載の画像処理装置。

【請求項 7】 画像データを記憶する第 1 の記憶手段と、符号データを記憶する第 2 の記憶手段と、設定された処理手順に応じて画像データを符号化し符号データを復号する処理手段と、前記処理手段に異なる処理手順を順次設定する制御手段とを備え、前記処理手段は、前記制御手段によって設定された処理手順に応じて、順次、前記第 1 の記憶手段に記憶された画像データを符号化し、前記第 2 の記憶手段に記憶された符号データを復号することを特徴とする画像処理装置。

【請求項 8】 前記制御手段は符号化と復号とを画像データの所定単位毎に切替えることを特徴とする請求項 7 に記載の画像処理装置。

【請求項 9】 前記処理手段はそれぞれ設定された処理手順を実行する直列に接続された少なくとも二つの処理部を有し、前記制御手段は一連の処理手順を分割して前記処理部のそれぞれに設定することを特徴とする請求項 5 から請求項 8 の何れかに記載の画像処理装置。

【請求項 10】 前記処理手順の分割は前記処理部それぞれの負荷が略均等になるように行うことを特徴とする請求項 9 に記載の画像処理装置。

【請求項 11】 前記処理部の間のデータのやり取りは

メモリを介して行うことを特徴とする請求項 9 に記載の画像処理装置。

【請求項 12】 入力画像データを複数のプロセッサを用いてブロック単位で符号化する画像処理方法であって、余白ブロックの画像データを符号化した結果得られる符号データを第 1 のメモリに記憶し、注目ブロックの前ブロックが余白ブロックであり、かつ該注目ブロックが余白ブロックである場合に、前記複数のプロセッサのうち一つが前記符号データを第 2 のメモリに書込むことを特徴とする画像処理方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は画像処理方法およびその装置に関し、例えば、画像データを符号化/復号する画像通信装置などの画像処理装置に関するものである。

【0002】

【従来の技術】膨大な情報量である画像データを伝送するには、その圧縮技術が必要不可欠である。画像データの符号化方法としては、グループ 3 ファクシミリで行われている MR 符号化方式やグループ 4 ファクシミリで行われている MMR 符号化方式などの白黒画像データの符号化方式、JPEG (Joint Photographic Experts Group) で標準化が進められている符号化方式（以下「JPEG 方式」という）や JBIG (Joint Bi-level Image Experts Group) で標準化が進められている符号化方式（以下「JBIG 方式」という）などのカラー画像データの符号化方式がある。これらの符号化方式は、符号化処理する画像データの性質、復号画像品質および符号化効率（圧縮率）に応じて、最適なものが適用される。

【0003】図 1 から図 4 は従来の画像処理装置における符号化/復号部の構成を示すブロック図である。従来の画像処理装置においては、白黒画像を MR 符号化あるいは MMR 符号化する場合は、図 1 に示すように二値符号化部 2 が構成され、カラー画像を JPEG 方式あるいは JBIG 方式で符号化する場合は、図 2 に示すように多値符号化部 3 とが構成された。また、白黒画像の符号化とカラー画像の符号化とを同時に行う場合は、図 3 に示すように、それぞれの符号化方式の符号化部を構成することで同時処理を可能にしていた。

【0004】さらに、同一の符号化方式による符号化と復号あるいは異なる符号化方式による符号化と復号を交互に行う場合は、図 4 に示すように、符号化と復号をそれぞれ行う専用部 5 および 6 を設けていた。

【0005】

【発明が解決しようとする課題】しかし、上記従来例においては、次のような問題点があった。すなわち、従来の画像処理装置において、複数の符号化方式を適用する場合は、適用する符号化方式毎に専用の処理部を設けなければならない欠点があった。また、符号化と復号とを

行う場合も、それぞれ専用の処理部、つまり符号化部と復号部とが必要になる欠点があった。その場合、符号化/復号部の周辺回路を含めたスペースやコスト、さらに複数の符号化方式を適用した場合にはその周辺回路や各符号化/復号部の制御の複雑化が問題になった。

【0006】

【課題を解決するための手段】本発明は、前記の課題を解決することを目的としたもので、前記の課題を解決する一手段として、以下の構成を備える。すなわち、設定された処理手順に応じて画像データを異なる符号化方式で順次符号化する画像処理方法にする。

【0007】また、画像データを記憶する記憶手段と、設定された処理手順に応じて画像データを符号化する処理手段と、前記処理手段に異なる符号化方式の処理手順を順次設定する制御手段とを備え、前記処理手段は、前記制御手段によって設定された処理手順に応じて、前記記憶手段に記憶された画像データを異なる符号化方式で順次符号化する画像処理装置にする。

【0008】また、設定された処理手順に応じて、順次、画像データを符号化し、符号データを復号する画像処理方法とする。また、画像データを記憶する第1の記憶手段と、符号データを記憶する第2の記憶手段と、設定された処理手順に応じて画像データを符号化し符号データを復号する処理手段と、前記処理手段に異なる処理手順を順次設定する制御手段とを備え、前記処理手段は、前記制御手段によって設定された処理手順に応じて、順次、前記第1の記憶手段に記憶された画像データを符号化し、前記第2の記憶手段に記憶された符号データを復号する画像処理装置にする。

【0009】

【作用】以上の構成によれば、設定された処理手順に応じて、画像データを異なる符号化方式で順次符号化する画像処理方法およびその装置を提供でき、例えば、複数の符号化方式を適用する場合でも、適用する符号化方式毎に専用の処理部を設ける必要がなく、専用の処理部を設けるためのスペースやコストを低減することができる。

【0010】また、設定された処理手順に応じて、順次、画像データを符号化し、符号データを復号する画像処理方法およびその装置を提供でき、例えば、符号化と復号とを行う場合でも、それぞれ専用の処理部を必要とせず、専用の処理部を設けるためのスペースやコストを低減し、制御を容易にすることができる。

【0011】

【実施例】以下、本発明にかかる一実施例の画像処理装置を図面を参照して詳細に説明する。

【0012】

【第1実施例】図6は本発明にかかる一実施例の画像処理装置を備える画像通信装置の構成例を示すブロック図である。同図において、101は原稿読取部で、イメー

ジスキャナなどにより原稿の画像データを読込む。なお、以下の説明においてはバンドスキャン方式とし、符号化する画像データおよび復号される画像データは、図7に一例を示すよう副走査方向に例えば128ライン幅のシャトル単位とする。

【0013】102は画像メモリで、例えばRAMなどで構成され、原稿読取部101により読込んだ画像データあるいは復号された画像データを格納する。103は画像出力部で、画像メモリ102に格納された画像データの画像を表示または印刷するディスプレイまたはプリンタである。104はDSPで、インストラクションメモリ108に格納された処理プログラムと、ローカルメモリ104に格納された係数などによって、画像データの符号化あるいは符号データの復号を行う。

【0014】105は符号メモリで、例えばRAMなどで構成され、DSP104により符号化された符号データあるいは復号する符号データを格納する。106は回線制御部（以下「CCU」という）で、符号メモリ105に格納された符号データを伝送路110に送出し、伝送路110から受信した符号データを符号メモリ105へ出力する。

【0015】109はマイクロプロセッサ（以下「CPU」という）で、例えばROMやRAMを内蔵していて、ROMに格納されたプログラムに応じて、画像メモリ102、DSP104、符号メモリ105などを制御し、上述したその他の構成にも不図示の信号線やバスなどを介して制御信号を送る。また、CPU109は、DSP104の処理に必要な係数などをローカルメモリ107に設定し、DSP104の処理プログラムをインストラクションメモリ108にダウンロードする。なお、ローカルメモリ107およびインストラクションメモリ108はRAMなどで構成する。

【0016】なお、画像出力部103にはストレージデバイスも備えることもでき、読取った画像データや復号した画像データなどを格納して、必要ときに符号化したり表示したりすることもできる。さらに、読取った画像データを符号化した符号データや、伝送路110から受信した符号データも格納でき、CPU109の制御によって、符号データを符号メモリ105へ転送して送信したり復号することもできる。ストレージデバイスにはハードディスク、フロッピーディスク、光磁気ディスクなどを使用する。

【0017】図8および図9は本実施例の処理手順の一例を示すフローチャートである。本実施例が起動されると、CPU109は、イニシャル処理を行うプログラムをインストラクションメモリ108にダウンロードする。また、ローカルメモリ107に設定する係数などがある場合は、同時にローカルメモリ107の設定も行う。DSP104は、ステップS101でイニシャル処理プログラムのダウンロードを監視し、ダウンロードが終了する

と、ステップS102でイニシャル処理を実行する。

【0018】次に、CPU109は、第1の符号化方式に対応する処理プログラムをインストラクションメモリ108にダウンロードを開始すると同時に、制御信号を送って原稿読取部101に原稿の読取りを開始させる。原稿読取部101から出力された1スキャン分の画像データは順次画像メモリ102に格納される。一方、DSP104は、ステップS103でこの処理プログラムのダウンロードを監視し、ダウンロードが終了すると、画像メモリ102に格納された画像データの符号化を開始する。つまり、ステップS104で画像メモリ102から所定単位で画像データを読み込み、ステップS105でインストラクションメモリ108に格納されたプログラム（第1の符号化方式）により符号化を実行し、ステップS106で符号データを符号メモリ105へ書込む。

【0019】このとき、画像メモリ102に格納された画像データが多値画像データであり、インストラクションメモリ108に格納された処理プログラムの符号化方式が二値画像用のものであれば、DSP104は、画像メモリ102から読込んだ画像データを二値化してから符号化を行う。また、この二値化のしきい値として、ローカルメモリ107に設定された係数などを用いることもできる。

【0020】次に、DSP104は、ステップS107で、1シャトルの画像データの符号化が終了したか否かを判定して、未了であればステップS104へ処理を戻す。つまり、第1のシャトルのすべての画像データが符号化されるまでステップS104からS107が繰返される。第1のシャトルの画像データのすべてが第1の符号化方式で符号化されると、DSP104は、第1の符号化方式による符号化が終了したことをCPU109へ伝え、ステップS108で、CPU109によって第2の符号化方式の処理プログラムがダウンロードされるのを待つ。

【0021】このダウンロードが終了すると、DSP104は、第1のシャトルの画像データに対して、第2の符号化方式による符号化処理を施す。つまり、ステップS109で画像メモリ102から所定単位で画像データを読み込み、ステップS110でインストラクションメモリ108に格納されたプログラム（第2の符号化方式）により符号化を実行し、ステップS111で符号データを符号メモリ105へ書込む。なお、ステップS111においては、第2の符号化方式による符号データは、第1の符号化方式による符号データが格納された領域とは別の領域に格納される。つまり、第1の符号化方式による符号データと、第2の符号化方式による符号データとは、符号メモリ105の異なる領域に格納されることになる。

【0022】次に、DSP104は、ステップS112で、1シャトルの画像データの符号化が終了したか否か

を判定して、未了であればステップS109へ処理を戻す。つまり、1シャトルのすべての画像データが符号化されるまでステップS109からS112が繰返される。第1のシャトルの画像データのすべてが第2の符号化方式で符号化されると、DSP104は、ステップS113で、原稿画像すべての符号化が終了したか否かを判定して、終了していれば符号化スタンバイ状態になり、未了であればステップS114で次のシャトルの画像データをCPU109へ要求する。この要求を受けたCPU109は、制御信号を送って原稿読取部101に次のシャトルの読取りを開始させる。DSP104は、ステップS115で画像メモリ102に次のシャトルの画像データが格納されるのを監視し、画像データが格納されるとステップS103へ戻って、第1の符号化方式の処理プログラムがダウンロードされるのを待ち、ダウンロードが終了すると、第2のシャトルの画像データに対して前述と同様の手順で符号化を行う。

【0023】このようにして、DSP104は、原稿画像全体の符号化が終了するまで前述した処理を繰返し、全体の符号化が終了すると符号化スタンバイ状態になる。なお、スタンバイ状態において、CPU109から再び符号化の指示があると、ステップS103から処理を再開する。符号メモリ105に格納された二つの符号化方式による符号データは、CPU109の制御および選択によってCCU106を介して伝送路110へ送信される。このCPU109による符号データの選択は、符号データ量や、通信相手に対応する符号化方式などに応じて行われる。また、シャトル毎の画像の特徴、例えば二値画像と多値画像とに応じて選択してもよい。

【0024】以上説明したように、本実施例によれば、インストラクションメモリ108にダウンロードする処理プログラムをシャトル単位で変えることにより、一つの画像を異なる符号化方式で符号化することができる。上述した第1、第2の符号化方式には、例えば、二値画像圧縮に適したMMR方式や所謂JBIGの動的算術符号化方式と、多値画像圧縮に適した所謂JPEGのADCT方式とを当てはめることができる。また組み合わせは任意に行うことができる。

【0025】

【第2実施例】以下、本発明にかかる第2実施例の画像処理装置を説明する。なお、第2実施例において、第1実施例と略同様の構成については、同一符号を付して、その詳細説明を省略する。図10および図11は本実施例の処理手順の一例を示すフローチャートである。

【0026】本実施例が起動されると、CPU109は、イニシャル処理を行うプログラムをインストラクションメモリ108にダウンロードする。また、ローカルメモリ107に設定する係数などがある場合は、同時にローカルメモリ107の設定も行う。DSP104は、ステップS201でイニシャル処理プログラムのダウンロード

を監視し、ダウンロードが終了すると、ステップS202でイニシャル処理を実行する。

【0027】次に、CPU109は、符号化プログラムをインストラクションメモリ108にダウンロードを開始すると同時に、制御信号を送って原稿読取部101に原稿の読取りを開始させる。原稿読取部101から出力された1スキャン分の画像データは順次画像メモリ102に格納される。一方、DSP104は、ステップS203でこの符号化プログラムのダウンロードを監視し、ダウンロードが終了すると、画像メモリ102に格納された画像データの符号化を開始する。つまり、ステップS204で画像メモリ102から所定単位で画像データを読込み、ステップS205でインストラクションメモリ108に格納された符号化プログラムにより符号化を実行し、ステップS206で符号データを符号メモリ105へ書込む。

【0028】このとき、画像メモリ102に格納された画像データが多値画像データであり、インストラクションメモリ108に格納された符号化プログラムが二値画像用のものであれば、DSP104は、画像メモリ102から読込んだ画像データを二値化してから符号化を行う。また、この二値化のしきい値として、ローカルメモリ107に設定された係数などを用いることもできる。

【0029】次に、DSP104は、ステップS207で、1シャトルの画像データの符号化が終了したか否かを判定して、未了であればステップS204へ処理を戻す。つまり、第1のシャトルのすべての画像データが符号化されるまでステップS204からS207が繰返される。第1のシャトルのすべての画像データが符号化されると符号化を中断し、符号化あるいは復号のスタンバイ状態になる。

【0030】続いて、DSP104は、ステップS208で、伝送路110から受信した符号データが符号メモリ105にあるか否か、つまり復号されていない符号データがあるか否かを判定して、なければステップS214へ処理をジャンプさせ、あればステップS209へ処理を進める。一方、CPU109は、第1のシャトルの画像データの符号化が終了した際に、符号メモリ105に受信した符号データが格納されていると、その符号データの符号化方式に対応した復号プログラムをインストラクションメモリ108にダウンロードする。つまり、受信した符号データがJPEG方式で符号化されたものであれば同方式に準じた復号プログラムをダウンロードし、受信した符号データがMMR符号化方式で符号化されたものであれば同方式の復号プログラムをダウンロードする。なお、DSP104が符号化した符号データと、受信した符号データとは、符号メモリ105の異なる領域に格納されている。

【0031】DSP104は、ステップS209で、CPU109によって復号プログラムがダウンロードされるのを

待ち、このダウンロードが終了すると、符号メモリ105に格納された符号データの復号を開始する。つまり、ステップS210で符号メモリ105から符号データを読込み、ステップS211でインストラクションメモリ108に格納された復号プログラムにより復号を実行し、ステップS212で復号した画像データを画像メモリ102へ書込む。なお、復号した画像データと、符号化する画像データとは、画像メモリ102の異なる領域に格納される。

【0032】次に、DSP104は、ステップS213で、1シャトル分の符号データの復号が終了したか否かを判定して、未了であればステップS210へ処理を戻す。つまり、1シャトル分の画像データが復号されるまでステップS210からS213が繰返される。1シャトル分の画像データが復号された場合、または、ステップS208で復号されていない符号データがなかった場合、DSP104は、ステップS214で、原稿画像すべての符号化が終了したか否かを判定して、終了していればステップS216へ処理を進め、未了であればステップS215へ処理を進める。

【0033】原稿画像すべての符号化が未了の場合、CPU104は、ステップS215で次のシャトルの画像データをCPU109へ要求する。この要求を受けたCPU109は、制御信号を送って原稿読取部101に次のシャトルの読取りを開始させる。DSP104は、ステップS218で画像メモリ102に次のシャトルの画像データが格納されるのを監視し、画像データが格納されるとステップS203へ戻って、符号化プログラムがダウンロードされるのを待ち、ダウンロードが終了すると、第2のシャトルの画像データに対して前述と同様の手順で符号化を行う。なお、ステップS208で復号されていない符号データがなかった場合は、符号化プログラムは既にダウンロードされている。

【0034】他方、原稿画像すべての符号化が終了した場合、CPU104は、ステップS216で、すべての符号データの復号が終了したか否かを判定して、終了していればスタンバイ状態になり、未了であればステップS217で次のシャトル分の符号データをCPU109へ要求し、ステップS209へ戻って復号プログラムがダウンロードされるのを待ち、ダウンロードが終了すると、第2のシャトル分の画像データに対して前述と同様の手順で復号を行う。なお、ステップS214で原稿画像のすべてが符号化されていた場合は、復号プログラムは既にダウンロードされている。

【0035】このようにして、DSP104は、原稿画像全体の符号化が終了するまで前述した符号化を繰返し、受信した符号データの復号が終了するまで前述した復号を繰返して、全体の符号化および復号が終了するとスタンバイ状態になる。なお、符号化が復号よりも早く終了した場合は復号のみを繰返し行い、復号が符号化よりも

早く終了した場合は符号化のみを繰返し行う。また、スタンバイ状態において、CPU 109から再び符号化の指示があるとステップS203から処理を再開し、CPU 109から再び復号の指示があるとステップS209から処理を再開する。

【0036】符号メモリ105に格納された符号データは、CPU 109の制御によってCCU 106を介して伝送路110へ送信される。また、画像メモリ102に格納された画像データ、つまり復号された画像データは、画像出力部103に送られて画像が出力される。以上説明したように、本実施例によれば、インストラクションメモリ108にダウンロードする処理プログラムをシャトル単位（所定のブロック単位）で変えることにより、符号化と復号とを共通の処理回路により交互に行うことができる。このため、画像入力部としての原稿読取部101からの画像入力および符号化と、CCU 106を経由して受信した符号データの復号および画像出力部103への出力とを、時分割してシャトル単位で並行して行うことができる。

【0037】

【第3実施例】以下、本発明にかかる第3実施例の画像処理装置を説明する。なお、第3実施例において、第1実施例と略同様の構成については、同一符号を付して、その詳細説明を省略する。上述した第1実施例においては、図5に一例を示すように、符号化部をデジタルシグナルプロセッサ（以下「DSP」という）7で構成し、インストラクションメモリ8にダウンロードする処理プログラムを変更することによって、符号化方式を切替える画像処理装置の一例を説明した。しかし、このような画像処理装置は、適用する符号化方式の処理が複雑な場合や処理行程が多い場合には、符号化時間が長くなるという問題があり、本実施例はこれを解決するものである。

【0038】図12は本実施例の画像処理装置を備える画像通信装置構成例を示すブロック図である。同図において、104aはDSPで、CPU 109によってインストラクションメモリ107aに格納されたプログラムと、CPU 109によってローカルメモリ108aに格納された係数などに従って、画像メモリ102に格納された画像データ、または、画像メモリ102に書込む画像データの処理を行う。

【0039】104bもDSPで、CPU 109によってインストラクションメモリ107bに格納されたプログラムと、CPU 109によってローカルメモリ108bに格納された係数などに従って、画像メモリ102に格納された画像データ、または、画像メモリ102に書込む画像データの処理を行う。111はデュアルポートRAM（以下「DPRAM」という）で、DSP 104aとDSP 104bとデータのやり取りを仲介するために、両DSPの間に配置されCPU 109によって制御される。

【0040】また、114と115はそれぞれ同期信号で、同期信号114はDSP 104aからDSP 104bへ送られ、同期信号115はDSP 104bからDSP 104aへ送られるものである。図13および図14はDSP 104aの処理手順の一例を、図15および図16はDSP 104bの処理手順の一例をそれぞれ示すフローチャートで、詳細は後述するが、同期信号114と115により同期をとられて実行されるものである。

【0041】本実施例が起動されると、CPU 109は、
10 イニシャル処理を行うプログラムをインストラクションメモリ108a、108bにダウンロードする。また、ローカルメモリ107a、107bに設定する係数などがある場合は、同時にローカルメモリ107a、107bの設定も行う。DSP 104a、104bはそれぞれ、ステップS301、S401でイニシャル処理プログラムのダウンロードを監視し、ダウンロードが終了すると、ステップS302、S402でイニシャル処理を実行する。

【0042】次に、CPU 109は、第1の符号化方式に
20 対応する処理プログラムのインストラクションメモリ108a、108bへのダウンロードを開始する。符号化する画像がカラーの場合は、例えば、上述したJPEG方式またはJBIG方式の処理プログラムの前半部分をインストラクションメモリ108aに、後半部分をインストラクションメモリ108bにダウンロードする。なお、例えば、前半部分の処理としては色空間変換と離散コサイン変換（以下「DCT」という）などの直交変換、後半部分の処理としては量子化とハフマン符号化などである。また、符号化する画像が白黒の場合は、例えば、MR符号化方式またはMMR符号化方式の処理プログラムの前半部分をインストラクションメモリ108aに、後半部分をインストラクションメモリ108bにダウンロードする。なお、例えば、前半部分の処理としては多値画像を二値画像に変換する二値化、後半部分の処理としては符号化などである。

【0043】つまり、ダウンロードされた二つの処理プログラムにより、画像データに一方式の符号化を施すようにする。従って、分割した各処理は、その負荷が同程度になるのが望ましい。これは復号処理を分割するときも同様である。CPU 109は、処理プログラムのダウンロードとともに、制御信号を送って原稿読取部101に原稿の読取りを開始させる。原稿読取部101から出力された1スキャン分の画像データは順次画像メモリ102に格納される。

【0044】[DSP 104aの処理手順] DSP 104aは、ステップS303で処理プログラムのダウンロードを監視し、ダウンロードが終了すると、ステップS115で画像メモリ102に次のシャトルの画像データが格納されるのを監視し、画像データが格納されると、画像メモリ102に格納された画像データの処理を開始す
50

る。つまり、ステップS305で画像メモリ102から所定単位で画像データを読み込み、ステップS306で同期信号115を判定してDPRAM111への書込みが可能になるのを待ち、可能になるとステップS307でインストラクションメモリ108aに格納されたプログラムにより符号化処理を実行し、ステップS308で処理データをDPRAM111へ書込む。

【0045】続いて、DSP104aは、ステップS309で同期信号114を反転した後、ステップS310で1シャトルの画像データの処理が終了したか否かを判定して、未了であればステップS305へ処理を戻す。つまり、第1のシャトルのすべての画像データが処理されるまでステップS305からS310が繰返される。1シャトルの画像データの処理が終了すると、DSP104aは、ステップS311で、現在の処理が第2の符号化方式か否かを判定して、そうであればステップS313へ処理を進める。また、そうでなければステップS312で、第2の符号化方式の処理プログラムのダウンロードをCPU109へ要求した後、ステップS303へ処理を戻し、ステップS303からS310で第2の符号化方式により前述と同様の処理を実行して、第1のシャトルのすべての画像データに第2の符号化方式の処理を施す。

【0046】また、現在の処理が第2の符号化方式であった場合、DSP104aは、ステップS313で、原稿画像すべての処理が終了したか否かを判定して、終了していればスタンバイ状態になり、未了であれば、ステップS314で第1の符号化方式の処理プログラムのダウンロードをCPU109に要求し、ステップS315で次のシャトルの画像データをCPU109へ要求する。この要求を受けたCPU109は、第1の符号化方式の処理プログラムをインストラクションメモリ107aにダウンロードするとともに、制御信号を送って原稿読取部101に次のシャトルの読取りを開始させる。この要求を発した後、DSP104aは、ステップS303へ処理を戻し、ダウンロードおよび第2のシャトルの画像データの設定が終了すると、ステップS305からS310で第1の符号化方式により前述と同様の処理を実行して、第2のシャトルのすべての画像データに第1の符号化方式の処理を施す。

【0047】このようにして、DSP104aは、原稿画像全体の処理が終了するまで前述した処理を繰返し、全体の処理が終了するとスタンバイ状態になる。なお、スタンバイ状態において、CPU109から再び処理の指示があると、ステップS303から処理を再開する。

【DSP104bの処理手順】 DSP104bは、ステップS403で処理プログラムのダウンロードを監視し、ダウンロードが終了すると、ステップS404で同期信号114を判定してDPRAM111からの読出しが可能になるのを待ち、読出し可能になると、DPRAM111に格納され

たデータの処理を開始する。つまり、ステップS405でDPRAM111から所定単位でデータを読み込み、ステップS406で同期信号115を反転し、ステップS407でインストラクションメモリ108bに格納されたプログラムにより符号化処理を実行し、ステップS408で符号データを符号メモリ105へ書込み、ステップS409で1シャトルの画像データの処理が終了したか否かを判定して、未了であればステップS404へ処理を戻す。つまり、第1のシャトルのすべての画像データが処理されるまでステップS404からS409が繰返される。

【0048】1シャトルの画像データの処理が終了すると、DSP104bは、ステップS410で、現在の処理が第2の符号化方式か否かを判定して、そうであればステップS412へ処理を進める。また、そうでなければステップS411で、第2の符号化方式の処理プログラムのダウンロードをCPU109へ要求した後、ステップS403へ処理を戻し、ステップS403からS410で第2の符号化方式により前述と同様の処理を実行して、第1のシャトルのすべての画像データに第2の符号化方式の処理を施す。

【0049】また、現在の処理が第2の符号化方式であった場合、DSP104bは、ステップS412で、原稿画像すべての処理が終了したか否かを判定して、終了していればスタンバイ状態になり、未了であれば、ステップS413で第1の符号化方式の処理プログラムのダウンロードをCPU109に要求する。この要求を受けたCPU109は、第1の符号化方式の処理プログラムをインストラクションメモリ107bにダウンロードする。この要求を発した後、DSP104bは、ステップS403へ処理を戻し、ダウンロードが終了すると、ステップS404からS409で第1の符号化方式により前述と同様の処理を実行して、第2のシャトルのすべての画像データに第1の符号化方式の処理を施す。

【0050】【同期方法】両DSPは、同期信号114と115の状態に基づいて、DPRAM111の書込みおよび読出しを実行する。つまり、符号化時において、DSP104aは両同期信号の状態が異なる場合DPRAM111への書込みを行い、DSP104bは両同期信号の状態が一致する場合DPRAM111からの読込みを行う。

【0051】図17はDSP104aと104bとの同期関係を説明するタイミングチャートである。同図において、601および613のイニシャル処理で、同期信号114がLレベルに、同期信号115がHレベルにそれぞれ設定されたとすると、両同期信号の状態は異なるので、DSP104aは、603で画像メモリ102から画像データを読み込み、604で符号化処理を行い、605でDPRAM111へ処理データを書込んで、同期信号114を反転するので、606で再び画像データを読込んだ後、607でスタンバイ状態になる。

【0052】一方、DSP104bは、最初、DPRAM111から読出すことはできないので、614でスタンバイ状態になる。同期信号114が反転すると、615でDPRAM111から処理データを読込んで、同期信号115を反転し、616で符号化処理を行い、617で符号メモリ105へ符号データを書込んだ後、618でスタンバイ状態になる。

【0053】同期信号115が反転すると、DSP104aは、608で符号化処理を行い、609でDPRAM111へ処理データを書込んで、同期信号114を反転するので、610で再び画像データを読込んだ後、611でスタンバイ状態になる。同期信号114が反転すると、DSP104bは、619で再びDPRAM111から処理データを読込んで、同期信号115を反転する。

【0054】このようにして両DSPは処理のタイミングをとりながら、原稿画像全体の符号化が終了するまで前述した符号化を繰返し、全体の符号化が終了するとスタンバイ状態になる。なお、スタンバイ状態において、CPU109から再び符号化の指示があると、DSP104aはステップS303から処理を再開し、DSP104bはステップS403から処理を再開する。

【0055】符号メモリ105に格納された符号データは、CPU109の制御によってCCU106を介して伝送路110へ送信される。以上説明したように、本実施例によれば、第1実施例と同様の効果があるほか、二つのDSPで符号化処理を分担するので、適用する符号化方式の処理が複雑な場合や処理行程が多い場合でも、符号化時間が長くなることはない。さらに、二つのDSP間にDPRAMを接続したので、DSPは処理したデータその内部メモリに蓄えておく必要がないなど、データのやり取りを容易にする効果もある。

【0056】

【第4実施例】以下、本発明にかかる第4実施例の画像処理装置を説明する。なお、第4実施例において、第3実施例と略同様の構成については、同一符号を付して、その詳細説明を省略する。上述した第2実施例においては、処理プログラムを変更することによって符号化/復号を切替える図5に一例を示す構成の画像処理装置を説明した。しかし、このような画像処理装置は、適用する符号化方式の処理が複雑な場合や処理行程が多い場合には、符号化/復号時間が長くなるという問題があり、本実施例はこれを解決するものである。

【0057】図18はDSP104bの処理手順の一例を、図19はDSP104aの処理手順の一例をそれぞれ示すフローチャートである。なお、基本となる処理手順は第3実施例で説明したものと同一なので、図18と図19には復号に関する部分のみを示した。つまり、図18は第3実施例のステップS410とS412（図16）の間に、図19は第3実施例のステップS311とS313（図14）の間にそれぞれ挿入されるものであ

る。

【0058】[DSP104bの処理手順] 画像メモリ102に格納された1シャトルの画像データが二つの符号化方式で符号化されると、DSP104bは、ステップS421で、伝送路110から受信した符号データが符号メモリ105にあるか否か、つまり復号されていない符号データがあるか否かを判定して、なければステップS412へ処理をジャンプさせ、あればステップS422へ処理を進める。

【0059】一方、CPU109は、1シャトルの画像データの符号化が終了した際に、符号メモリ105に受信した符号データが格納されていると、その符号データの符号化方式に対応した復号処理プログラムをインストラクションメモリ108bにダウンロードする。つまり、受信した符号データがJPEG方式で符号化されたものであれば同方式に準じた復号処理プログラムをダウンロードし、受信した符号データがMMR符号化方式で符号化されたものであれば同方式の復号処理プログラムをダウンロードする。なお、DSP104bが符号化した符号データと、受信した符号データとは、符号メモリ105の異なる領域に格納されている。

【0060】DSP104bは、ステップS422で、CPU109によって復号処理プログラムがダウンロードされるのを待ち、このダウンロードが終了すると、符号メモリ105に格納された符号データの復号を開始する。つまり、ステップS423で符号メモリ105から符号データを読み込み、ステップS424で同期信号114を判定してDPRAM111への書き込みが可能になるのを待ち、可能になるとステップS425でインストラクションメモリ108bに格納された復号処理プログラムにより復号処理を実行し、ステップS426で処理データをDPRAM111へ書き込む。なお、復号処理したデータと、符号化処理したデータとは、DPRAM111の異なる領域に格納される。

【0061】続いて、DSP104bは、ステップS427で同期信号115を反転した後、ステップS428で1シャトル分の処理が終了したか否かを判定して、終了であればステップS412へ処理を進め、未了であればステップS423へ処理を戻す。つまり、1シャトル分の符号データが処理されるまでステップS423からS428が繰返される。

【0062】[DSP104aの処理手順] 画像メモリ102に格納された1シャトルの画像データが二つの符号化方式で符号化されると、DSP104aは、ステップS321で、復号されていない処理データがDPRAM111にあるか否かを判定して、なければステップS313へ処理をジャンプさせ、あればステップS322へ処理を進める。

【0063】一方、CPU109は、1シャトルの画像データの符号化が終了した際に、DPRAM111に復号され

ていない処理データが格納されていると、その符号化方式に対応した復号処理プログラムをインストラクションメモリ108aにダウンロードする。DSP104aは、ステップS322で、CPU109によって復号処理プログラムがダウンロードされるのを待ち、このダウンロードが終了すると、ステップS323で同期信号115を判定してDPRAM111からの読み込みが可能になるのを待ち、可能になるとDPRAM111に格納された処理データの復号処理を開始する。つまり、ステップS324でDPRAM111から処理データを読み込み、ステップS325で同期信号114を反転し、ステップS326でインストラクションメモリ108aに格納された復号処理プログラムにより処理を実行し、ステップS327で復号した画像データを画像メモリ102へ書き込む。なお、復号した画像データと、符号化する画像データとは、画像メモリ102の異なる領域に格納される。

【0064】続いて、DSP104aは、ステップS328で1シャトル分の復号処理が終了したか否かを判定して、終了であればステップS313へ処理を進め、未了であればステップS323へ処理を戻す。つまり、1シャトル分の符号データが処理されるまでステップS323からS328が繰返される。なお、図18と図19には詳細を示さなかったが、ステップS412およびステップS313ではそれぞれ、処理の終了状態に応じた分岐をする。つまり、符号化と復号ともに処理が終了した場合はスタンバイ状態になり、何れも終了していない場合および復号だけが終了した場合はステップS413およびS314へ処理は進み、符号化だけが終了して復号が終了していない場合はステップS423およびS323へ処理は進む。

【0065】以上説明したように、本実施例によれば、第2実施例と同様の効果があるほか、二つのDSPで符号化/復号処理を分担するので、適用する符号化方式の処理が複雑な場合や処理行程が多い場合でも、符号化/復号時間が長くなることはない。さらに、二つのDSP間にDPRAMを接続したので、DSPは処理したデータその内部メモリに蓄えておく必要がないなど、データのやり取りを容易にする効果もある。

【0066】

【第5実施例】以下、本発明にかかる第5実施例の画像処理装置を説明する。なお、第5実施例において、第3実施例と略同様の構成については、同一符号を付して、その詳細説明を省略する。前述した第3実施例においては、二つのDSP間に設けたDPRAMによって、両DSPがデータをやり取りする構成を示したが、図20に一例を示すように、本実施例はこのDPRAMを削除した画像処理装置を備えた画像通信装置である。

【0067】図21はDSP104aの処理手順の一例を、図22はDSP104bの処理手順の一例をそれぞれ示すフローチャートである。なお、基本となる処理手順

は第3実施例で説明したものと同一なので、図21と図22には本実施例に特徴的な部分のみを示した。つまり、図21に示すステップS501からS504は、第3実施例のステップS306からS309(図13,14)に替わるものであり、図22に示すステップS601からS605は、第3実施例のステップS404からS408(図15,16)に替わるものである。

【0068】[DSP104aの処理手順] DSP104aは、ステップS305で画像メモリ102から所定単位で画像データを読み込み、ステップS501で、インストラクションメモリ108aに格納されたプログラムにより符号化処理を実行し、一旦内蔵メモリなどに蓄える。ステップS502で同期信号115を判定して処理データの転送が可能になるのを待ち、転送可能になるとステップS503で同期信号114を反転した後、ステップS504で処理データをDSP104bへ送信する。なお、転送が可能か否かの判定を行う同期信号は、符号化処理する画像データの単位が偶数番目か奇数番目かによって、その状態が反転するようなものとする。また、データの転送は、予め転送単位として決めた所定データ数毎に、両同期信号を変化させることによって行う。

【0069】続いて、DSP104aは、ステップS310で1シャトルの画像データの処理が終了したか否かを判定して、未了であればステップS305へ処理を戻す。つまり、1シャトルのすべての画像データが処理されるまで上記の処理が繰返される。

【DSP104bの処理手順】 DSP104bは、ステップS403で処理プログラムのダウンロードを監視し、ダウンロードが終了すると、ステップS601で同期信号114を判定してデータの転送が可能になるのを待ち、転送可能になるとステップS602で、DSP104bから処理データを受信し、一旦内蔵メモリなどに蓄える。つまり、ステップS602とステップS504とは同期して実行されるものである。

【0070】続いて、DSP104bは、ステップS603でインストラクションメモリ108bに格納されたプログラムにより符号化処理を実行し、ステップS604で符号メモリ105に符号データを書込み、ステップS605で同期信号115を反転した後、ステップS409で1シャトルの画像データの処理が終了したか否かを判定して、未了であればステップS601へ処理を戻す。つまり、1シャトルのすべての画像データが処理されるまでステップS601からS409が繰返される。

【0071】以上説明したように、本実施例によれば、二つのDSPで符号化処理を分担するので、適用する符号化方式の処理が複雑な場合や処理行程が多い場合でも、符号化時間が長くなることはなく、DPRAMが使用しない分、第3実施例に比べて構成が簡単になりコストも低減できる。

【0072】

【第6実施例】以下、本発明にかかる第6実施例の画像処理装置を説明する。なお、第6実施例において、第3実施例と略同様の構成については、同一符号を付して、その詳細説明を省略する。前述した第3実施例においては、二つのDSP間に設けたDPRAMによって、両DSPがデータをやり取りする構成を示したが、図20に一例を示すように、本実施例はこのDPRAMを削除した構成を備えた画像通信装置である。

【0073】図23はDSP104bの処理手順の一例を、図24はDSP104aの処理手順の一例をそれぞれ示すフローチャートである。なお、基本となる処理手順は第3実施例で説明したものと同じなので、図23と図24には復号に関する部分のみを示した。つまり、図23は第3実施例のステップS410とS412（図16）の間に、図24は第3実施例のステップS311とS313（図14）の間にそれぞれ挿入されるものである。

【0074】[DSP104bの処理手順] 画像メモリ102に格納された1シャトルの画像データが二つの符号化方式で符号化されると、DSP104bは、ステップS431で、伝送路110から受信した符号データが符号メモリ105にあるか否か、つまり復号されていない符号データがあるか否かを判定して、なければステップS412へ処理をジャンプさせ、あればステップS432へ処理を進める。

【0075】一方、CPU109は、第4実施例で説明したように、1シャトルの画像データの符号化が終了した際に、符号メモリ105に受信した符号データが格納されていると、その符号データの符号化方式に対応した復号処理プログラムをインストラクションメモリ108bにダウンロードする。DSP104bは、ステップS432で、CPU109によって復号処理プログラムがダウンロードされるのを待ち、このダウンロードが終了すると、符号メモリ105に格納された符号データの復号を開始する。つまり、ステップS433で符号メモリ105から符号データを読み込み、ステップS434でインストラクションメモリ108bに格納された復号処理プログラムにより復号処理を実行し、ステップS435で同期信号114を判定して処理データの転送が可能になるのを待ち、転送可能になるとステップS436で同期信号115を反転した後、ステップS437で処理データをDSP104aへ送信する。

【0076】続いて、DSP104bは、ステップS438で1シャトル分の処理が終了したか否かを判定して、終了であればステップS412へ処理を進め、未了であればステップS433へ処理を戻す。つまり、1シャトル分の符号データが処理されるまでステップS433からS438が繰返される。

【DSP104aの処理手順】画像メモリ102に格納された1シャトルの画像データが二つの符号化方式で符号

化されると、DSP104bは、ステップS331で、復号されていない処理データがDPRAM111にあるか否かを判定して、なければステップS313へ処理をジャンプさせ、あればステップS332へ処理を進める。

【0077】一方、CPU109は、第4実施例で説明したように、1シャトルの画像データの符号化が終了した際に、復号されていない処理データがあると、その符号化方式に対応した復号処理プログラムをインストラクションメモリ108aにダウンロードする。DSP104aは、ステップS332で、CPU109によって復号処理プログラムがダウンロードされるのを待ち、このダウンロードが終了すると、ステップS333で同期信号115を判定して処理データの読込みが可能になるのを待ち、可能になると処理データの復号処理を開始する。つまり、ステップS437と同期したステップS334で処理データを受信し、ステップS335でインストラクションメモリ108aに格納された復号処理プログラムにより処理を実行し、ステップS336で同期信号114を反転し、ステップS337で復号した画像データを画像メモリ102へ書込む。

【0078】続いて、DSP104aは、ステップS338で1シャトル分の復号処理が終了したか否かを判定して、終了であればステップS313へ処理を進め、未了であればステップS333へ処理を戻す。つまり、1シャトル分の符号データが処理されるまでステップS333からS338が繰返される。なお、図23と図24には詳細を示さなかったが、ステップS412およびステップS313ではそれぞれ、処理の終了状態に応じた分岐をする。つまり、符号化と復号ともに処理が終了した場合はスタンバイ状態になり、何れも終了していない場合および復号だけが終了した場合はステップS413およびS314へ処理は進み、符号化だけが終了して復号が終了していない場合はステップS433およびS333へ処理は進む。

【0079】以上説明したように、本実施例によれば、二つのDSPで符号化/復号処理を分担するので、適用する符号化方式の処理が複雑な場合や処理行程が多い場合でも、符号化/復号時間が長くなることはなく、DPRAMが使用しない分、第4実施例に比べて構成が簡単になりコストも低減できる。

【0080】

【第7実施例】以下、本発明にかかる第7実施例の画像処理装置を説明する。なお、第7実施例において、第3実施例と略同様の構成については、同一符号を付して、その詳細説明を省略する。上述した各実施例においては、JPEG方式を適用する場合は余白データの処理に時間がかかる問題があり、本実施例はこれを解決するものである。

【0081】まず、JPEG方式による画像データの符号化

成されるが、そのうち最も基本になるのがベースラインシステムである。図 25 はベースラインシステムのデータ圧縮手順を示す図である。まず、イメージスキャナなどの原稿読取装置で読取った入力画像データを、8×8画素からなるブロックに分割する。以後の圧縮処理はこのブロック単位で行う。JPEG方式では、原稿読取装置で読込んだ画像データの符号化処理する色空間に対する規定はないが、符号化処理に先立つ処理として色空間の変換処理が多く行われている。変換する色空間は、符号化処理を高能率で行うことが可能である色空間、例えば、輝度信号と色信号からなる色空間 YCbCr, YIQ, Lab などを使用することが多い。また多くの場合、色空間変換後の色信号に関してサブサンプリング処理を行っている。ベースラインシステムでは、分割された各ブロックの画像データ（イメージスキャナなどで読込んだ色空間あるいは色空間変換後の画像データ）に直交変換の一種である DCT を施す（401）。

【0082】DCTによって、入力画像データは空間周波数成分のデータに変換される。変換後の8×8係数の一番左上の係数を直流（DC）成分といい、変換前のブロックの画像データの平均値に相当する値になる。それ以外の63個の係数を交流（AC）成分といい、変換前のブロックの画像データにおいて、その位置に対応する空間周波数成分をどれだけ含んでいるかを示す。

【0083】8×8個のしきい値（以下「量子化テーブル」という）にスケールファクタ（以下「Qファクタ」という）を乗算した値で、変換後のDCT変換係数を除算して量子化する（402）。JPEG方式の場合、量子化は圧縮率を決める大きな要因になる。つまり、Qファクタを大きくすると、使用する量子化テーブルのしきい値が全体的に大きくなるため圧縮率は向上するが、画像情報をそれだけ抑圧するので画質は劣化する。逆に、Qファクタを小さくすると、使用する量子化テーブルのしきい値が全体的に小さくなるため圧縮率は悪くなるが、画質は向上する。JPEG方式のベースラインシステムでは、一頁の画像データの量子化に用いる量子化テーブルは輝度成分、色成分ともに一種類である。

【0084】量子化後のDCT係数には、DC成分とAC成分で異なる符号化処理を施す。DC成分は、隣接ブロック間の相関の強さを利用して、処理ブロックのDC成分と前8×8ブロックのDC成分との差分をハフマン符号化する（403、404）。AC成分は、ブロック内で空間周波数の低域から高域ヘジグザグスキャンを施し、一次元の配列に並べ直して、零以外の係数（有効係数）と、連続する零（無効係数）の個数（ランレングス）とを組にして、二次元のハフマン符号化を行う（405から409）。

【0085】さて、画像通信装置の符号化部は、符号化あるいは復号のほかに余白処理を行う必要がある。余白処理は、通信の際に必要な余白を送信側で付加する

ものであり、多くの場合、余白として付加する余白データは「白」である。従って、JPEG方式で符号化する場合、画像読取装置で読取った画像データだけでなく、付加した余白データも符号化する必要がある。

【0086】以下においては、説明を簡単にするため、本実施例はJPEG方式の符号化だけを実行するものとするが、第3実施例などと同様に、符号化方式を切替えて、一つの画像を異なる符号化方式で符号化することができるというまでもない。

10 【DSP104aの処理手順】図26と図27はDSP104aの処理手順の一例を示すフローチャートであり、第3実施例において図13と図14で説明した処理手順と略同様であるので、同ステップは同一符号で示し、各ステップの詳細説明は省略する。ただし、先述したように、本実施例は、JPEG方式の符号化だけしか実行しないので、第3実施例のステップS311、S312、S314はない。

20 【0087】【DSP104bの処理手順】図28から図30はDSP104bの処理手順の一例を示すフローチャートである。本実施例が起動されると、CPU109は、イニシャル処理を行うプログラムをインストラクションメモリ108bにダウンロードする。DSP104bは、ステップS701でイニシャル処理プログラムのダウンロードを監視し、ダウンロードが終了すると、ステップS702でイニシャル処理を実行する。

30 【0088】次に、CPU109は、JPEG方式に対応する処理プログラムのインストラクションメモリ108bへのダウンロードを開始する。DSP104bは、ステップS703で処理プログラムのダウンロードを監視し、ダウンロードが終了すると、ステップS704で同期信号115をDPRAM111の書込禁止に設定する。続いて、DSP104bは、ステップS705でローカルメモリ107bにマッピングした前ブロックデータフラグレジスタ（以下単に「レジスタ」という）2201を「1」にし、ステップS706でローカルメモリ107bにマッピングした余白コードメモリ2202に余白データの符号化後の値を設定する。なお、ローカルメモリ107bのマッピング状態例を図31に示す。

40 【0089】また、余白コードメモリ2202には、符号化するブロックデータが前ブロックのデータと同一であるとして、その符号データを設定する。符号化する信号が三信号であり、符号化処理形態がJPEG方式のブロックインタリーブである場合は、第1の信号のDC差分が零の符号、第1の信号のAC成分がすべて零（EOB: End of Block）の符号、第2の信号のDC差分が零の符号、第2の信号のAC成分がすべて零の符号（EOB）、第3の信号のDC差分が零の符号、第3の信号のAC成分がすべて零の符号（EOB）をつなげた符号である。符号化する色空間がYCbCrで、符号化時に参照するハフマンテーブルがJPG 50 G推奨テーブルの場合は、Y信号のDC差分が零の符号「0

0' と、Y信号のAC成分がすべて零の信号 '1010' と、Cb, CrのDC差分が零の信号 '00' と、Cb, Cr信号のAC成分がすべて零を示す符号 '00' とをつなげることになるので、余白コードメモリ 2202には '00101000000000' が設定される。

【0090】図32は符号化するブロック画像データの構成例を示す図であり、1401~1413で示すデータはJPEG方式で処理するブロックデータ (8×8画素) を、1414はレジスタ2201の初期値を、1415~1427は1401~1413を符号化した後のレジスタ2201の値をそれぞれ示し、また、1428~1440は、1401~1413の符号化時に、色空間変換、DCT、量子化、ハフマン符号化を行うか否かを、記号○ (行う) と記号× (行わない) で示している。本実施例においては、符号化するブロック画像データが余白データで、そのときのレジスタ2201の値が '1' の場合は、色空間変換、DCT、量子化、ハフマン符号化を行わない。また、データ1401は符号化するページの最初のデータであるとする。

【0091】DSP104bは、ステップS707で、符号化する画像データがページの最初のデータであるときはステップS715へ処理を進める。そして、ステップS715で、同ステップで符号化するデータが余白データか否かを判断するが、データ1401は余白データであるから、ステップS716で余白データの符号化を行う。つまり、余白データを1ブロック分 (三信号成分) 生成し、図25に示したブロックの処理に従い符号化する。なお、生成された符号データは順次符号メモリ105へ書込まれる。続いて、DSP104bは、ステップS723でレジスタ2201の値を '0' (図32の1415) にする。

【0092】送信する原稿全体の符号化が終了していれば、DSP104bは、ステップS714を経て、割込処理などによってCPU109へ符号化終了を知らせるとともにスタンバイ状態になるが、ここではデータ1402以降が存在するのでステップS708へ進む。DSP104bは、ステップS708で余白データであるか否かを判定して、データ1402は余白データであるからステップS709へ進む、ステップS709で前ブロックデータフラグの値を判定して、このときレジスタ2201は '0' (図32の1415) であるからステップS711へ進んで、ステップS711でレジスタ2201を '1' (図32の1416) にし、ステップS712で余白データを三信号成分生成し、ステップS713で生成した符号データを符号メモリ105へ書込む。

【0093】続くデータ1403も余白データであるからステップS714、S708を経てステップS709へ進むが、このときレジスタ2201は '1' (図32の1416) であるからステップS710へ進んで、ステップS710で、符号化は行わずに、余白コードメモ

リ2202に設定した符号データを符号メモリ105へ書込む。

【0094】続くデータ1404は余白データではないからステップS714、S708を経てステップS717へ進み、ステップS717で同期信号115をDPRAM111の書込許可に設定し、ステップS718で同期信号114が読込許可状態になるのを待つ。なお、この間にDSP104aは、処理データをDPRAM111へ書込み、同期信号114を反転する。読込許可状態になると、DSP104bは、ステップS719でDPRAM111から処理データを読込み、ステップS720で同期信号115を反転し、ステップS721で符号化処理を実行し、ステップS722で符号データを符号メモリ105へ書込む。

【0095】続いて、DSP104bは、ステップS723でレジスタ2201の値を '0' (図32の1418) にし、ステップS714で再びすべての画像データの符号化が終了したか否かを判定する。このようにして両DSPは処理のタイミングをとりながら、原稿画像全体の符号化が終了するまで前述した符号化を繰返し、全体の符号化が終了するとスタンバイ状態になる。なお、スタンバイ状態において、CPU109から再び符号化の指示があると、DSP104aはステップS303から処理を再開し、DSP104bはステップS703から処理を再開する。

【0096】符号メモリ105に格納された符号データは、CPU109の制御によってCCU106を介して伝送路110へ送信される。以上説明したように、本実施例によれば、第3実施例と同様の効果があるほか、JPEG方式によって画像データを符号化する場合において、余白データの処理を高速化する効果がある。

【0097】なお、上述した第1実施例から第7実施例においては、原稿読取部101が読取った画像データを格納する画像メモリ102と、符号データを格納する符号メモリ105とを別構成とする例を説明したが、同一メモリでもよいことはいふまでもない。また、一回に符号化または復号する画像データ量を1シャトル分としたが、2シャトル分以上にすることもできる。

【0098】また、上述した第1実施例、第3実施例、第5実施例においては、二つの符号化方式の例を説明したが、本実施例はこれに限定されるものではなく、三つ以上の符号化方式に対応させることもできる。また、上述した第2実施例、第4実施例、第6実施例においては、符号化と復号の符号化方式は同一であってもよいし、例えばJPEG方式とMMR符号化方式のように異なってもよく、符号化に関しては、第1実施例と同様に複数の符号化を行うこともできる。

【0099】上述した各実施例によれば、符号化/復号を行うDSPの処理プログラムを変えることにより、適用する符号化方式を変えることができるので、予め決めた

画像データ単位毎に複数の異なる符号化を実行できる。また、予め決めた画像データ単位毎に符号化と復号とを交互に行うことも可能になる。また、適用する符号化方式毎に専用の処理部を設ける必要がないので、符号化/復号部の周辺回路を含めたスペースやコストを最小限にすることができる。

【0100】さらに、処理プログラムを変えるだけで、複数の異なる符号化方式を適用することができるので、符号化/復号部およびその周辺回路の制御が容易になる。なお、本発明は、複数の機器から構成されるシステムに適用しても、一つの機器からなる装置に適用してもよい。また、本発明は、システムあるいは装置にプログラムを供給することによって達成される場合にも適用できることはいうまでもない。

【0101】

【発明の効果】以上、本発明によれば、設定された処理手順に応じて、画像データを異なる符号化方式で順次符号化する画像処理方法およびその装置を提供でき、例えば、複数の符号化方式を適用する場合でも、適用する符号化方式毎に専用の処理部を設ける必要がなく、専用の処理部を設けるためのスペースやコストを低減する効果がある。

【0102】また、設定された処理手順に応じて、順次、画像データを符号化し、符号データを復号する画像処理方法およびその装置を提供でき、例えば、符号化と復号とを行う場合でも、それぞれ専用の処理部を必要とせず、専用の処理部を設けるためのスペースやコストを低減し、制御を容易にする効果がある。

【図面の簡単な説明】

【図1】従来の画像処理装置における符号化の構成を示すブロック図である。

【図2】従来の画像処理装置における符号化の構成を示すブロック図である。

【図3】従来の画像処理装置における符号化の構成を示すブロック図である。

【図4】従来の画像処理装置における符号化部と復号部の構成を示すブロック図である。

【図5】本発明にかかる符号化方式を切替える画像処理装置の概略構成を示すブロック図である。

【図6】本発明にかかる一実施例の画像処理装置を備える画像通信装置の構成例を示すブロック図である。

【図7】バンドスキャン方式の一例を説明する図である。

【図8】本実施例の処理手順の一例を示すフローチャートである。

【図9】本実施例の処理手順の一例を示すフローチャートである。

【図10】本発明にかかる第2実施例の処理手順の一例を示すフローチャートである。

【図11】第2実施例の処理手順の一例を示すフローチャートである。

ャートである。

【図12】本発明にかかる第3実施例の画像処理装置を備える画像通信装置の構成例を示すブロック図である。

【図13】図12に示すDSP104aの処理手順の一例を示すフローチャートである。

【図14】図12に示すDSP104aの処理手順の一例を示すフローチャートである。

【図15】図12に示すDSP104bの処理手順の一例を示すフローチャートである。

10 【図16】図12に示すDSP104bの処理手順の一例を示すフローチャートである。

【図17】図12に示すDSP104aと104bとの同期関係を説明するタイミングチャートである。

【図18】本発明にかかる第4実施例のDSP104bの処理手順の一例を示すフローチャートである。

【図19】第4実施例のDSP104aの処理手順の一例を示すフローチャートである。

【図20】本発明にかかる第5実施例の画像処理装置を備える画像通信装置の構成例を示すブロック図である。

20 【図21】図20に示すDSP104aの処理手順の一例を示すフローチャートである。

【図22】図20に示すDSP104bの処理手順の一例を示すフローチャートである。

【図23】本発明にかかる第6実施例のDSP104bの処理手順の一例を示すフローチャートである。

【図24】第6実施例のDSP104aの処理手順の一例を示すフローチャートである。

【図25】JPEGベースラインシステムのデータ圧縮手順を示す図である。

30 【図26】本発明にかかる第7実施例のDSP104aの処理手順の一例を示すフローチャートである。

【図27】第7実施例のDSP104aの処理手順の一例を示すフローチャートである。

【図28】第7実施例のDSP104bの処理手順の一例を示すフローチャートである。

【図29】第7実施例のDSP104bの処理手順の一例を示すフローチャートである。

【図30】第7実施例のDSP104bの処理手順の一例を示すフローチャートである。

40 【図31】第7実施例のローカルメモリのマッピング状態例を示す図である。

【図32】符号化するブロック画像データの構成例を示す図である。

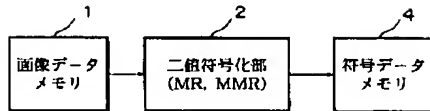
【符号の説明】

101	原稿読取部
102	画像メモリ
103	画像出力部
104	DSP
105	符号メモリ
50 106	回線制御部 (CCU)

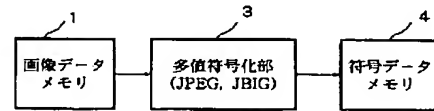
- 25
- 107 ローカルメモリ
108 インストラクションメモリ
109 マイクロプロセッサ (CPU)
111 デュアルポートRAM (DPRAM)

- 26
- 114, 115 同期信号
2201 前ブロックデータフラグレジスタ (レジスタ)
2202 余白コードメモリ

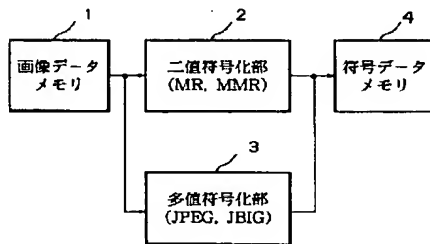
【図 1】



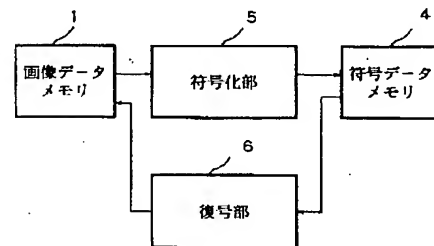
【図 2】



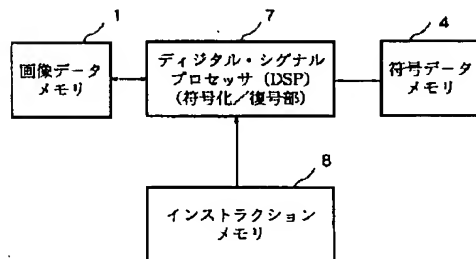
【図 3】



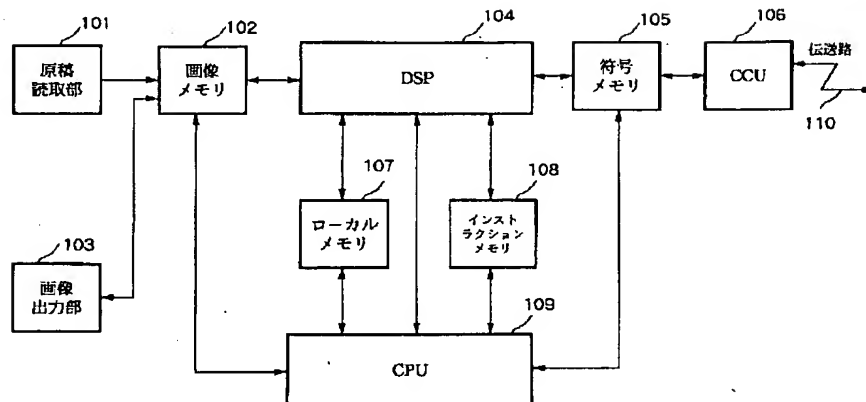
【図 4】



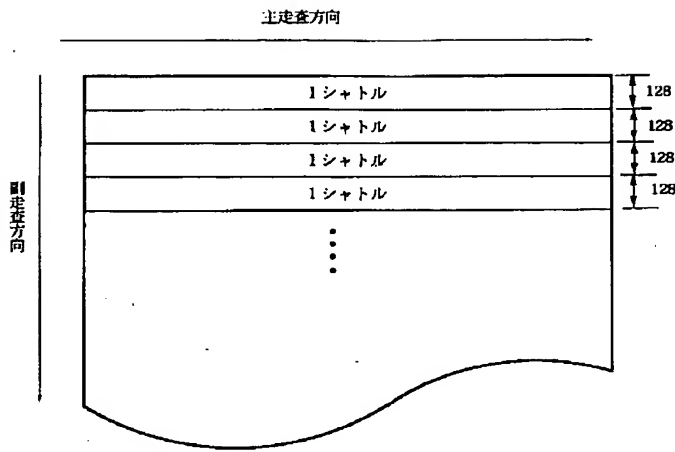
【図 5】



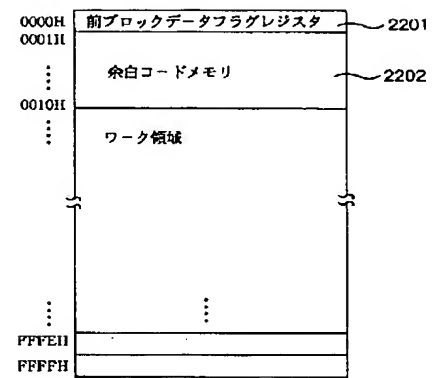
【図 6】



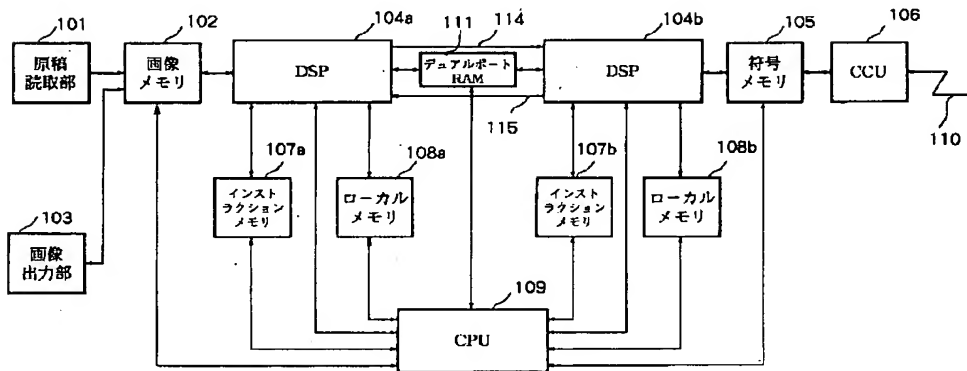
【図 7】



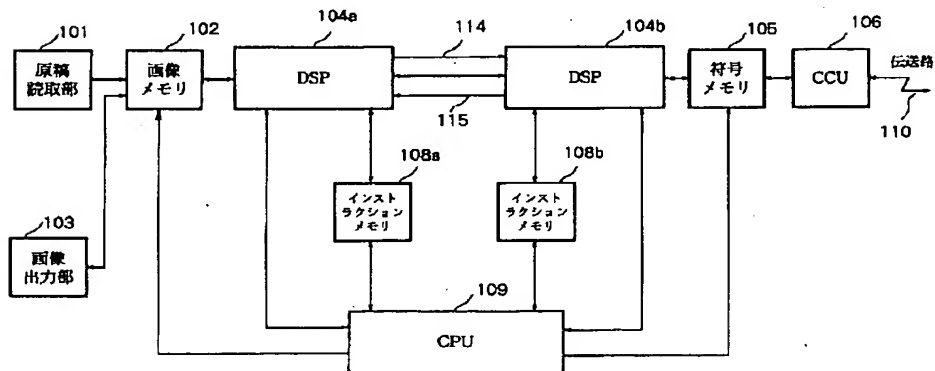
【図 31】



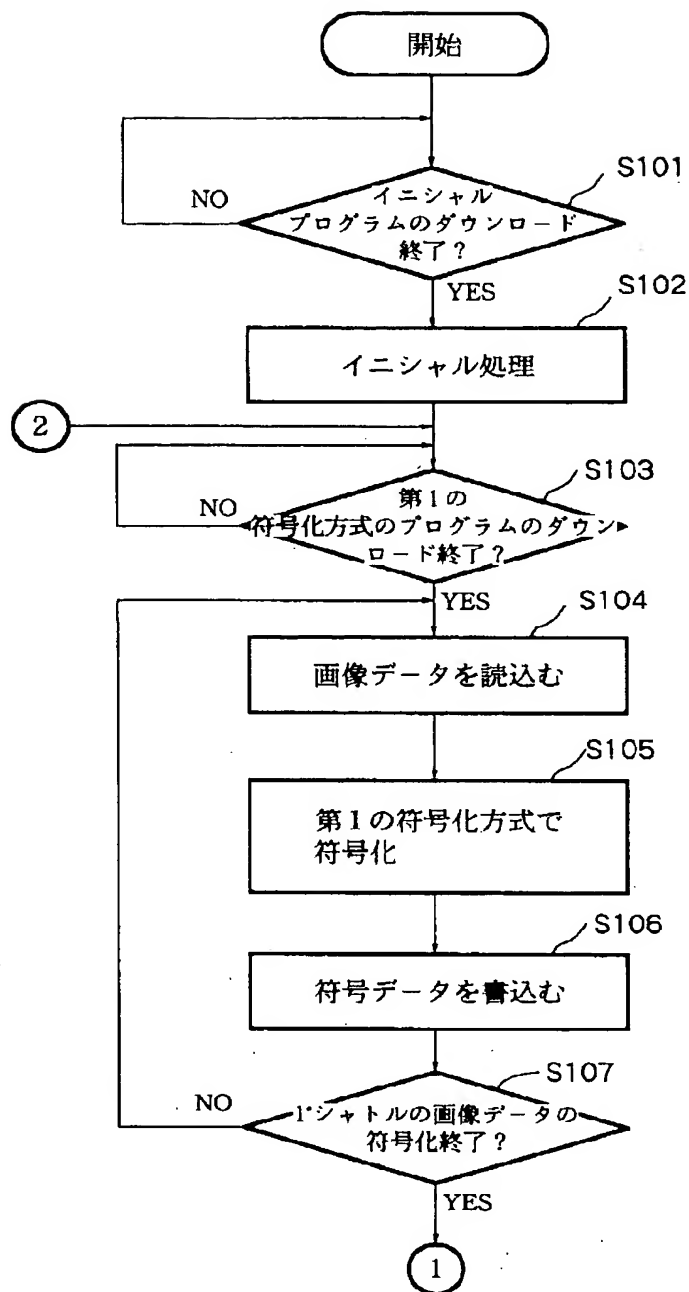
【図 12】



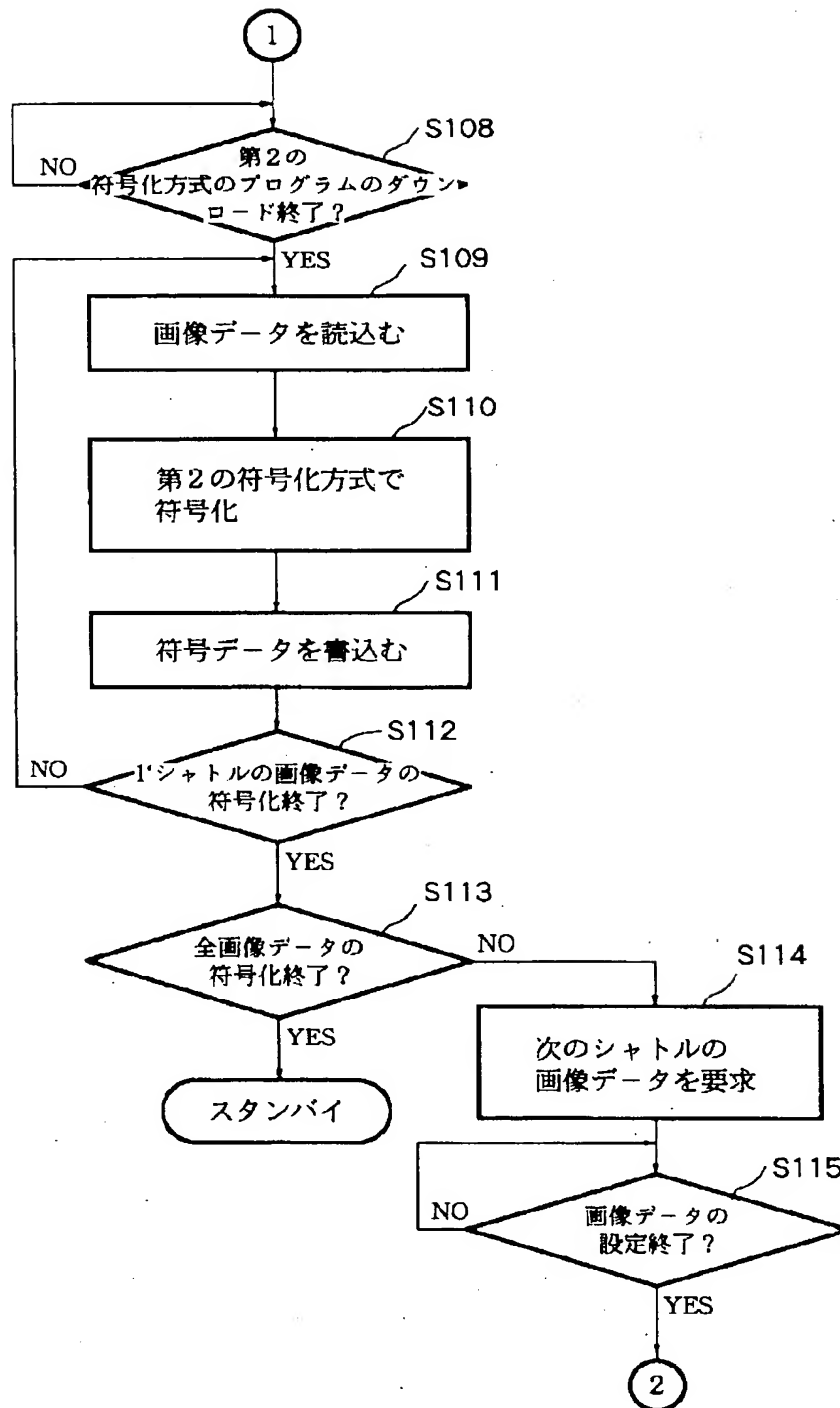
【図 20】



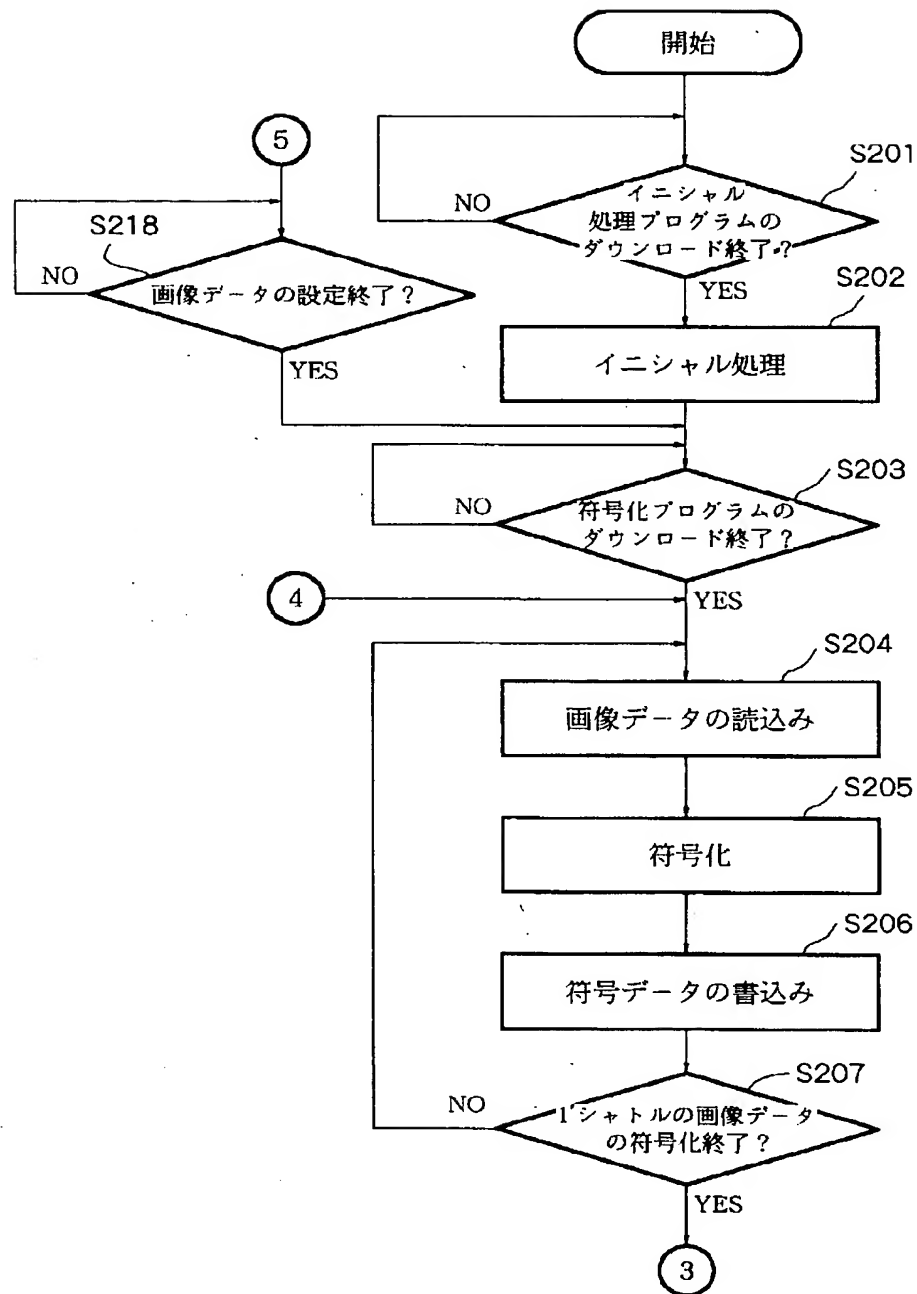
【图8】



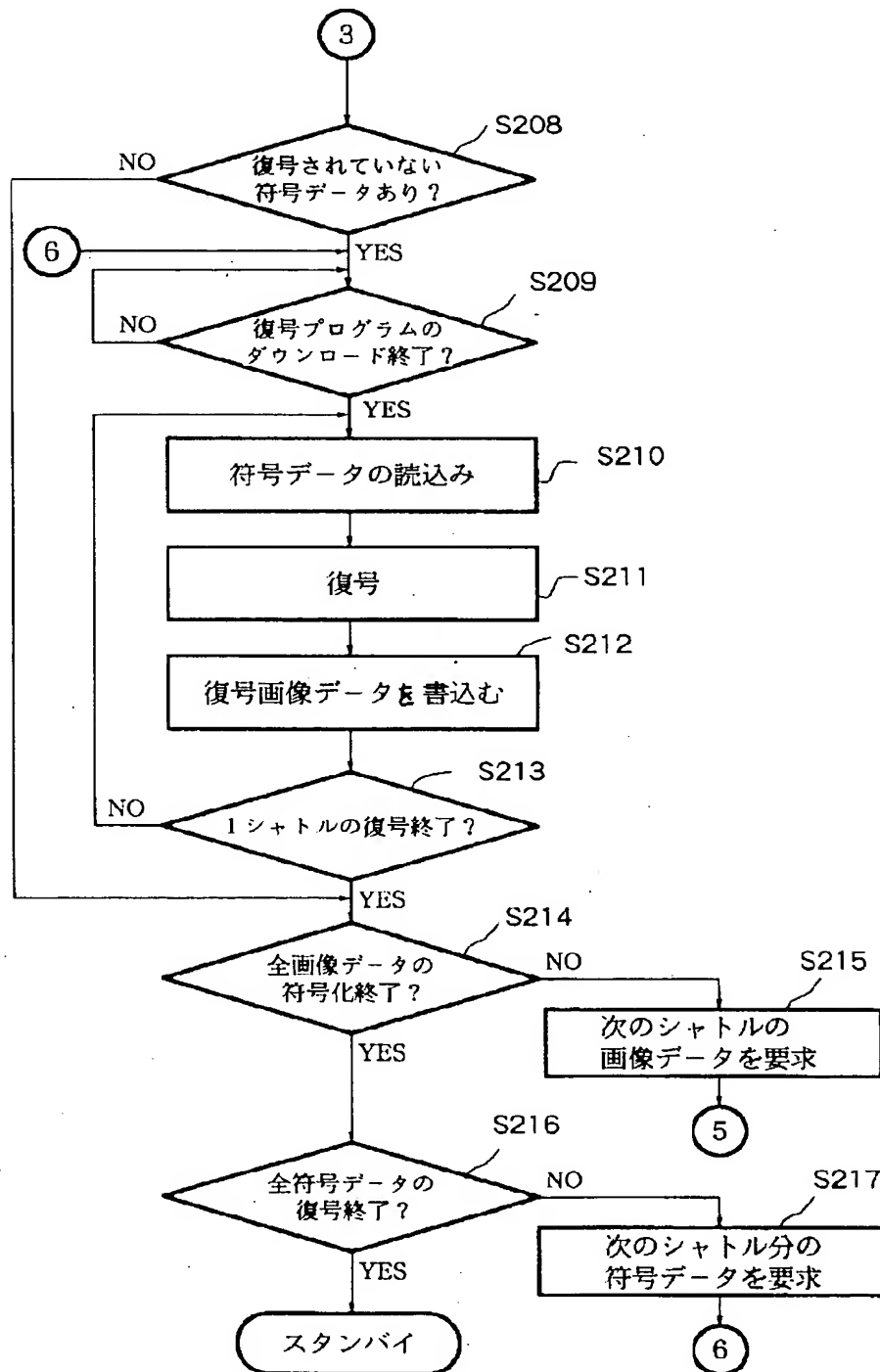
【図9】



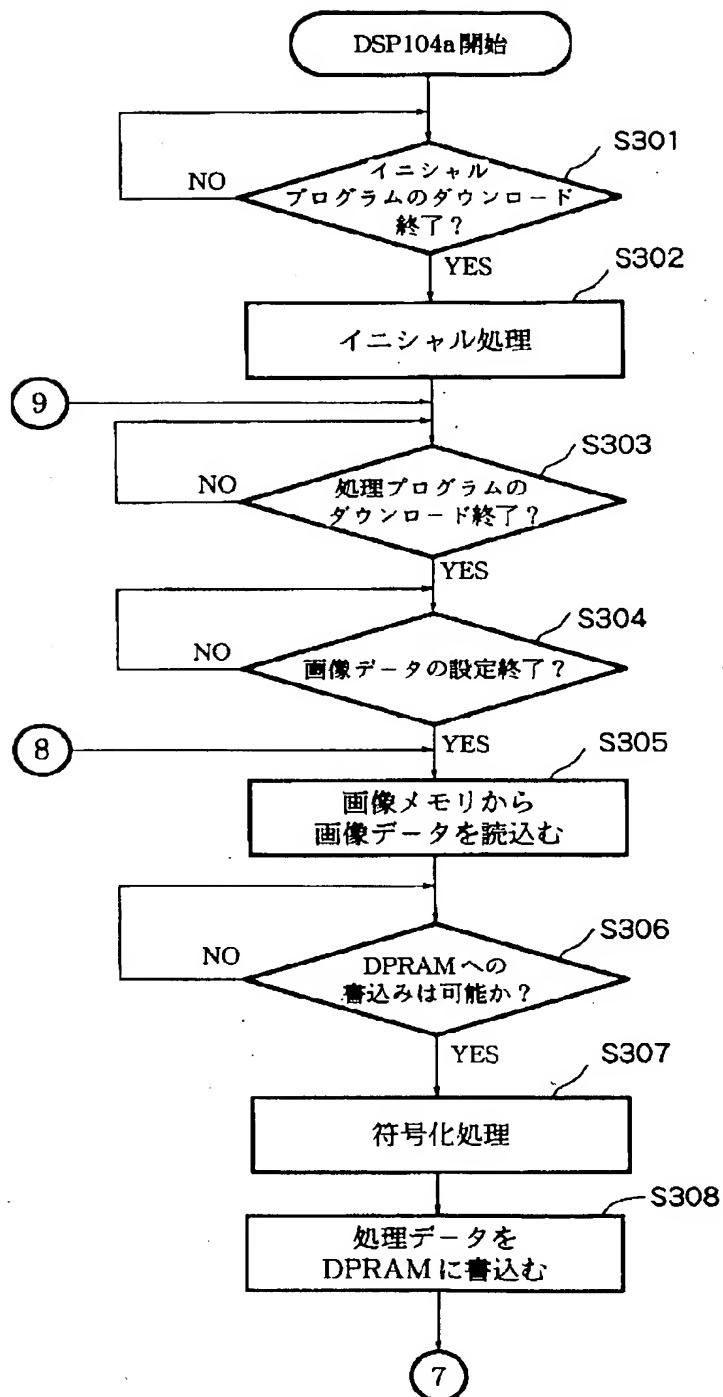
【図 10】



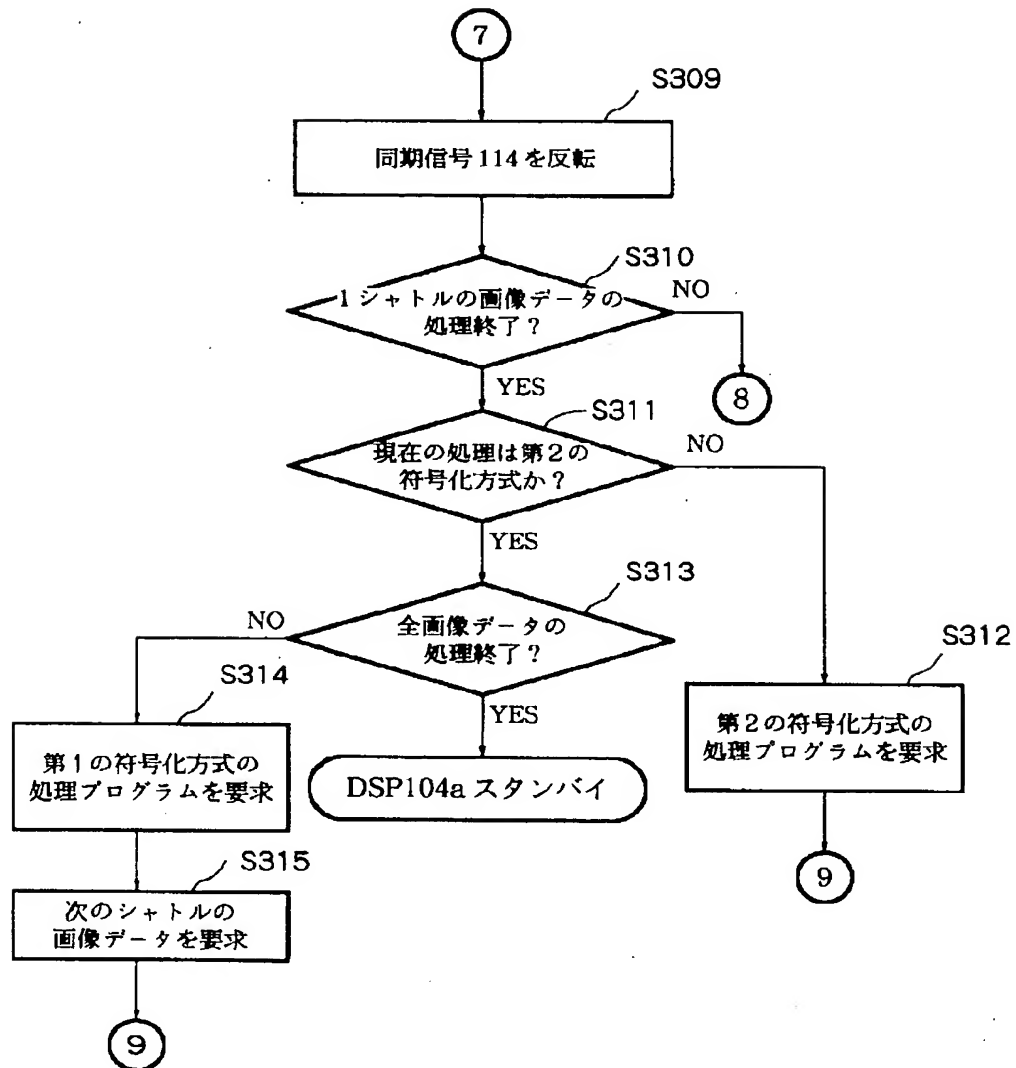
【図11】



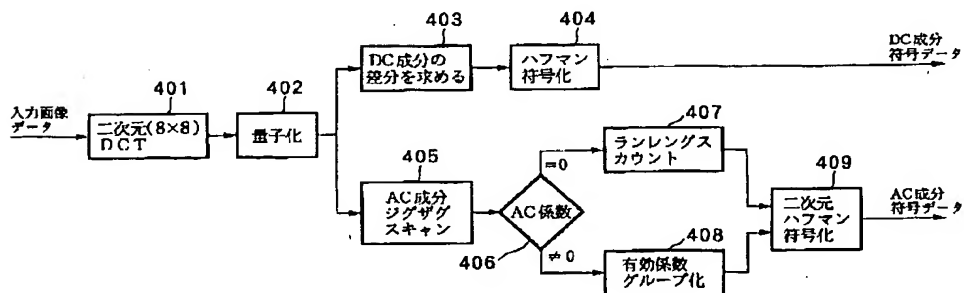
【図13】



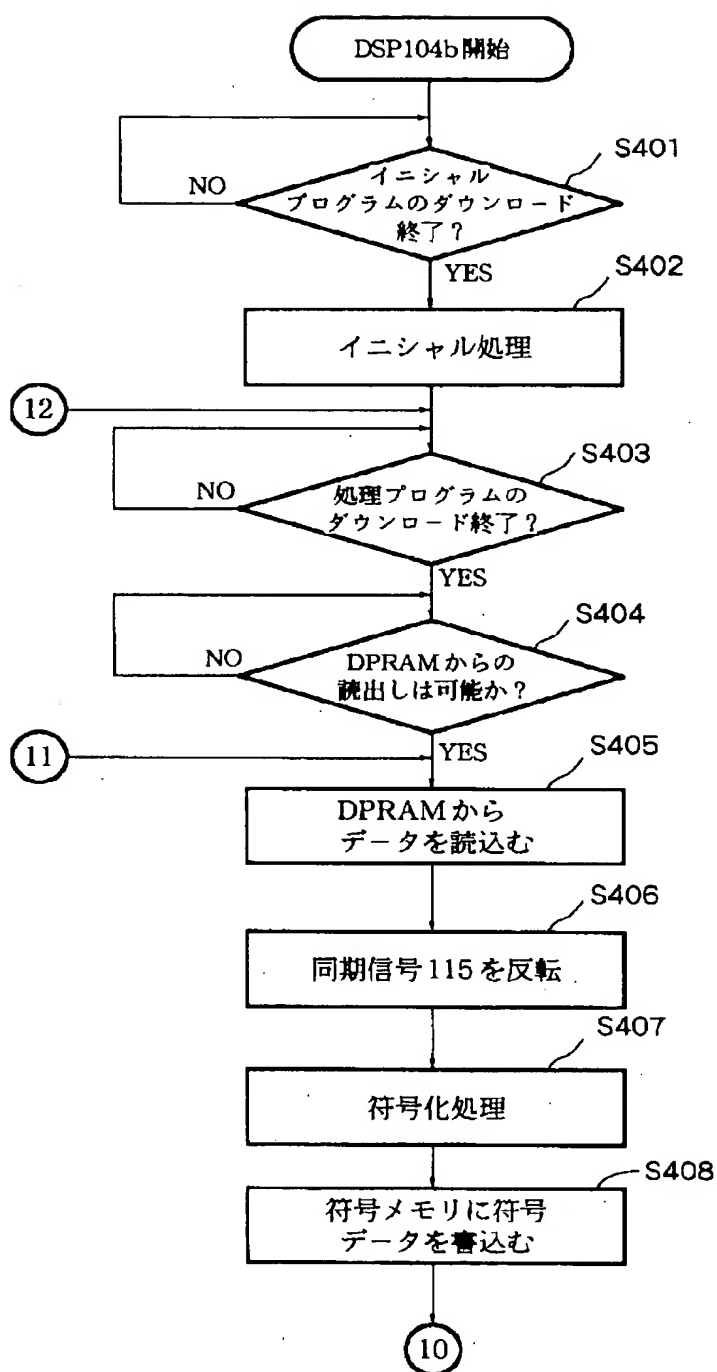
【図 14】



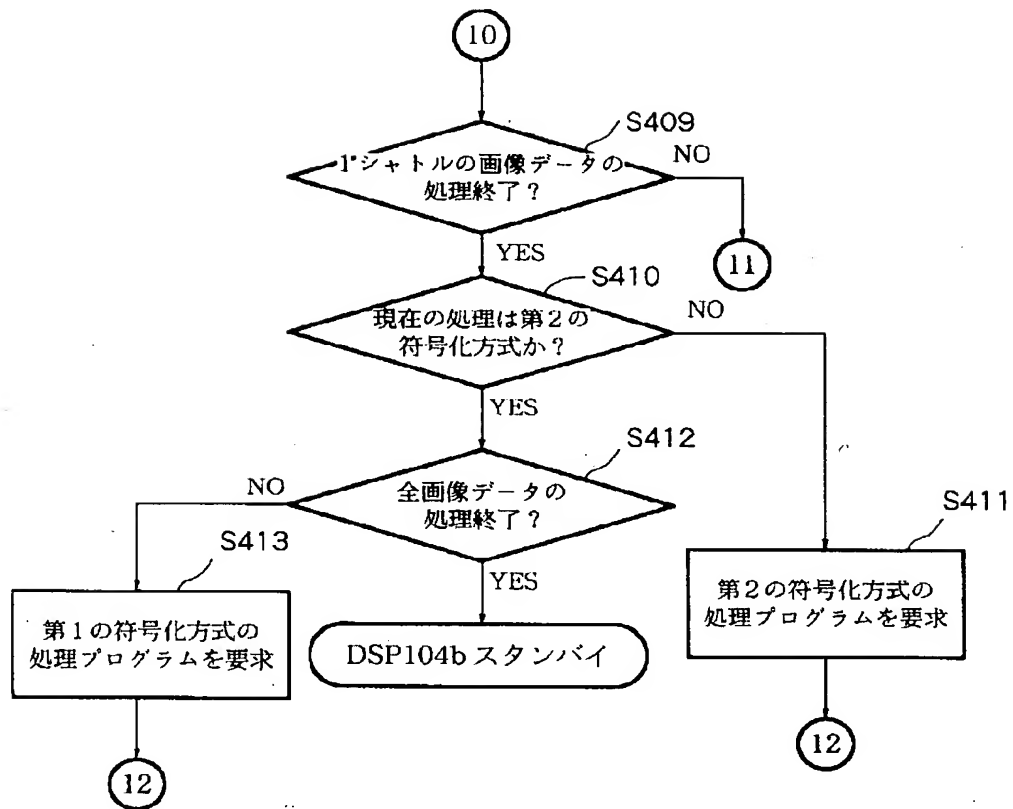
【図 25】



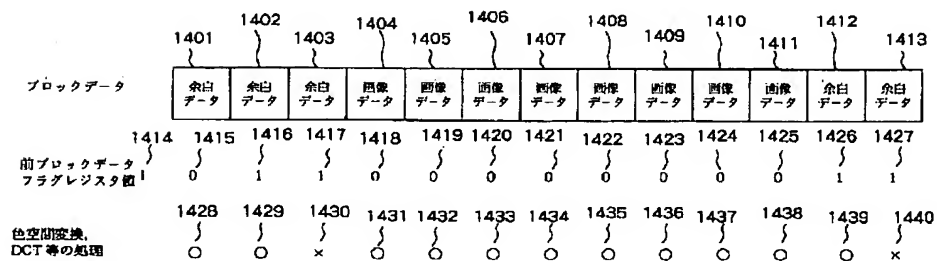
【図15】



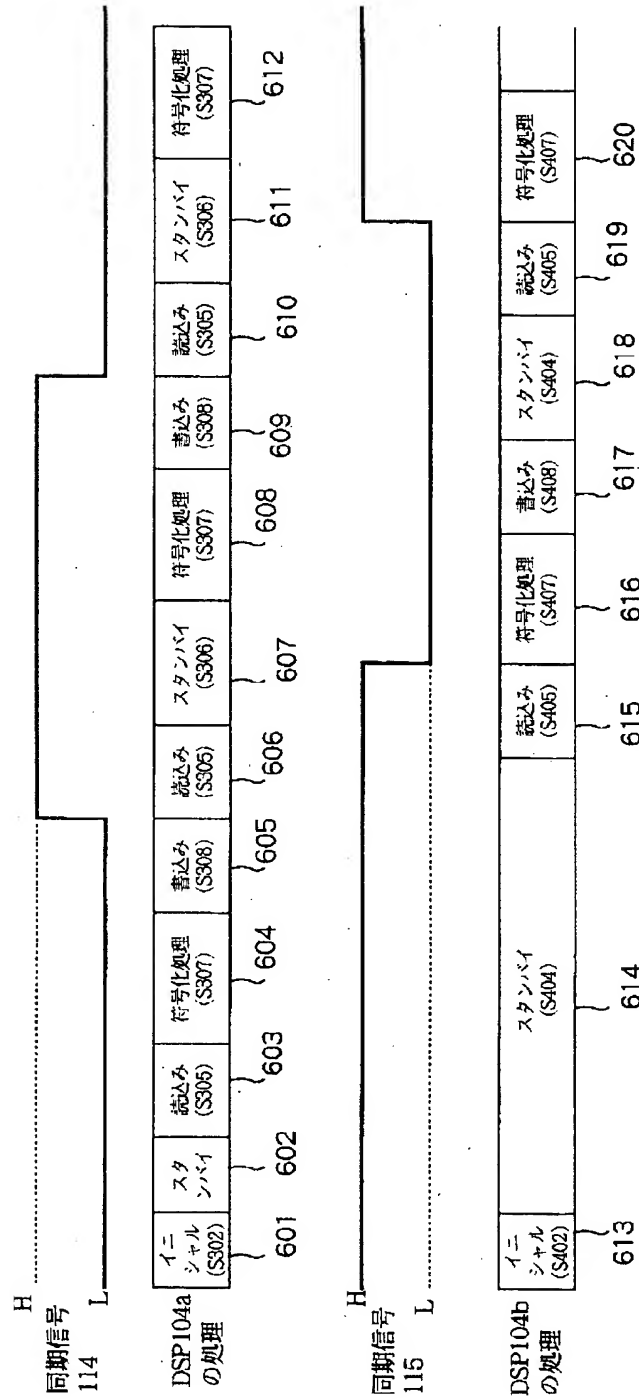
【図 16】



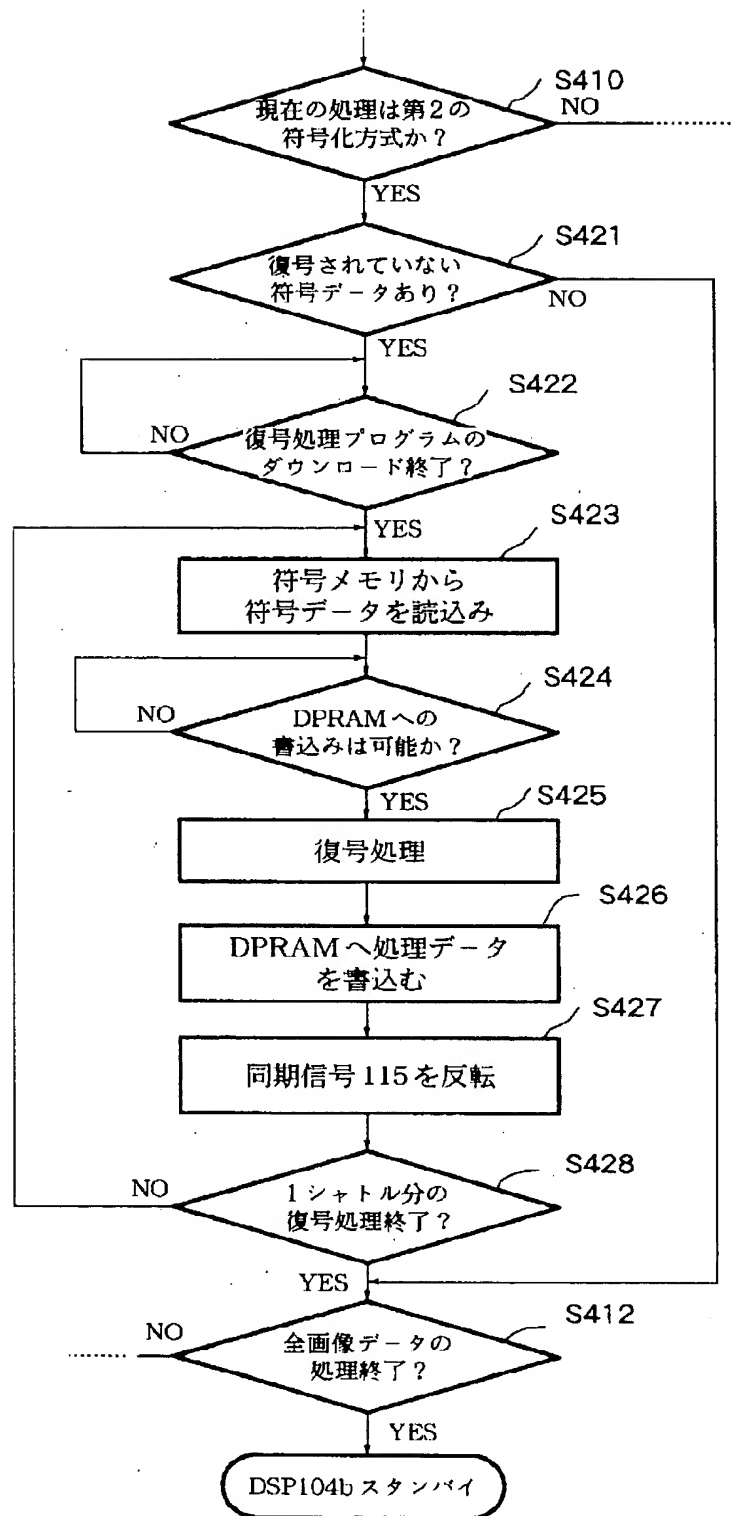
【図 32】



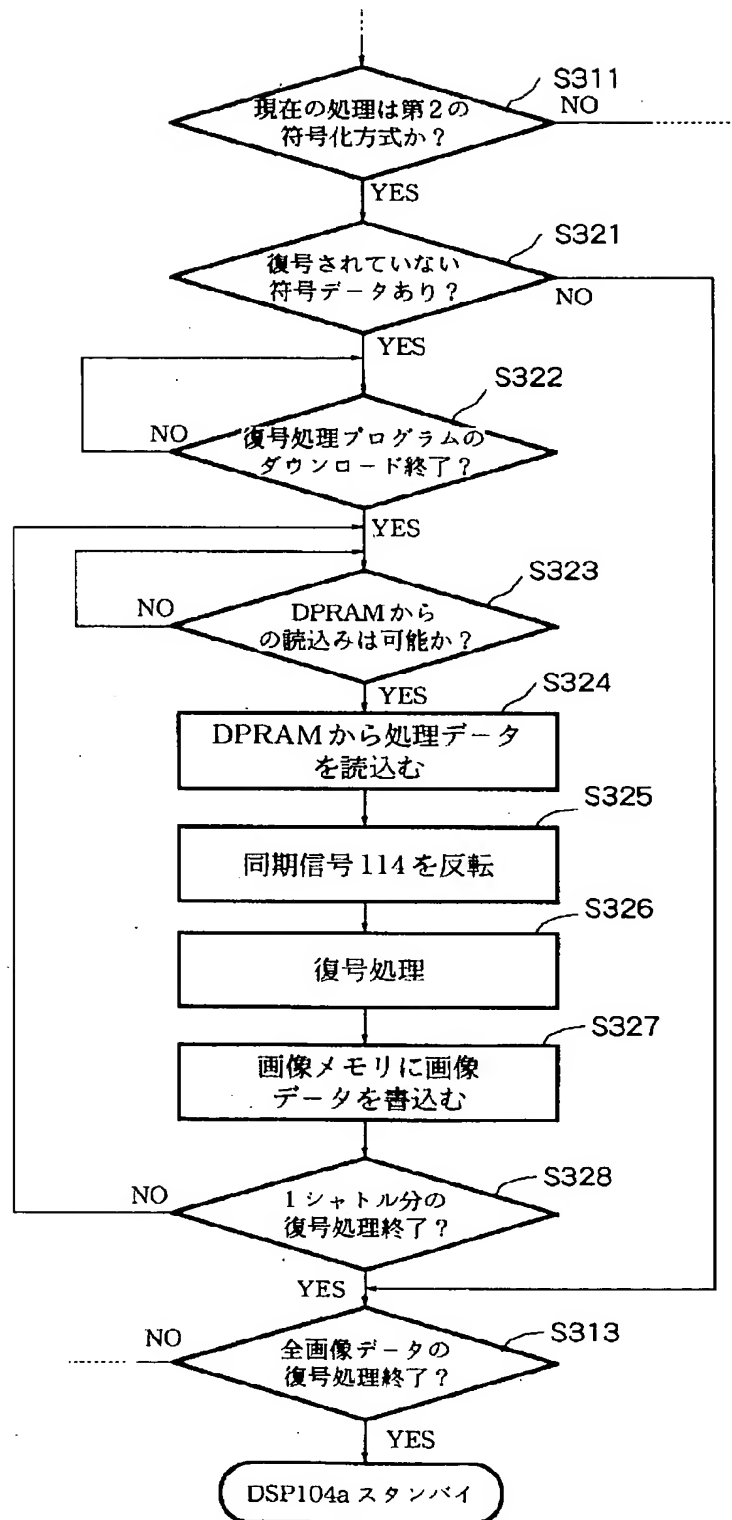
【図 17】



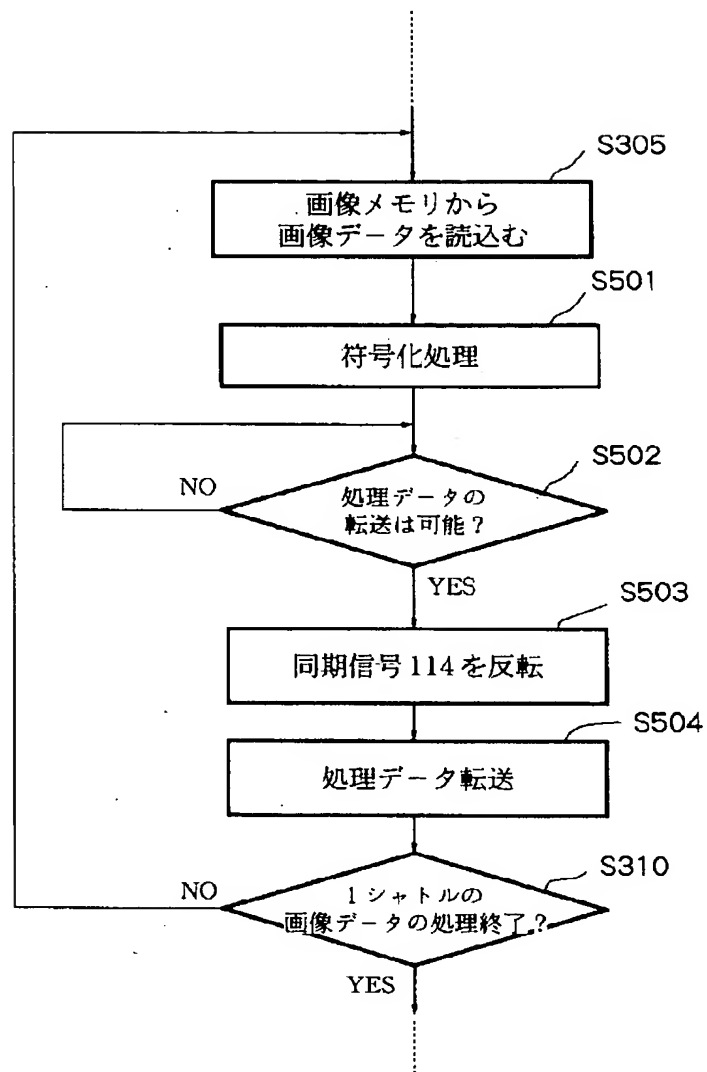
【図18】



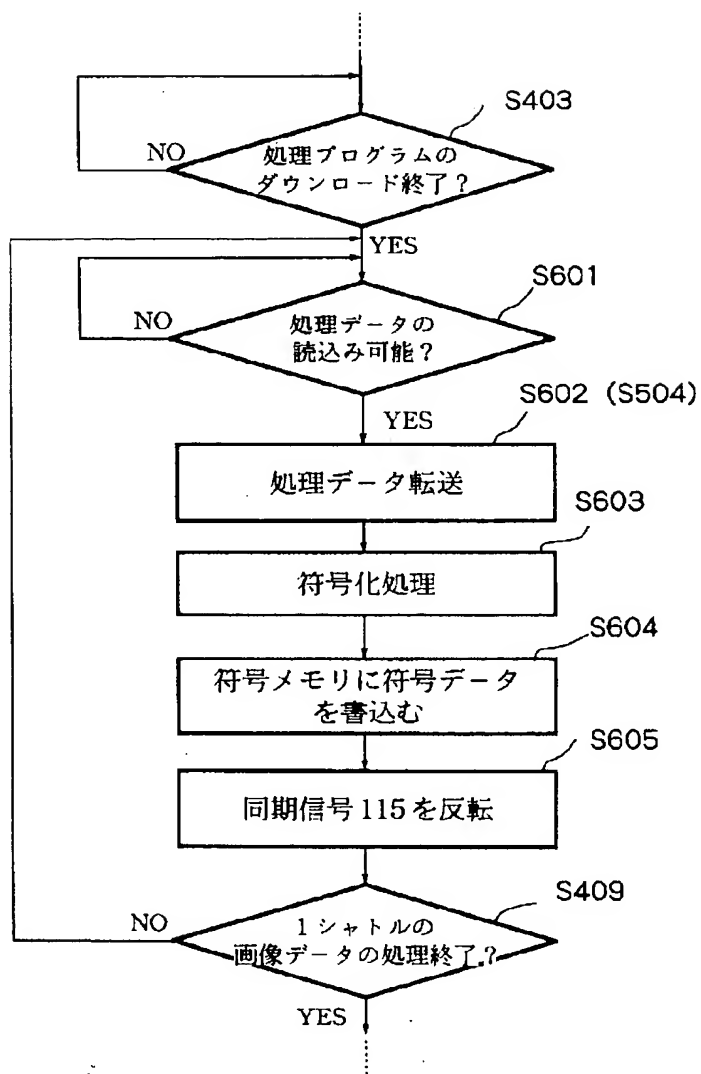
【図19】



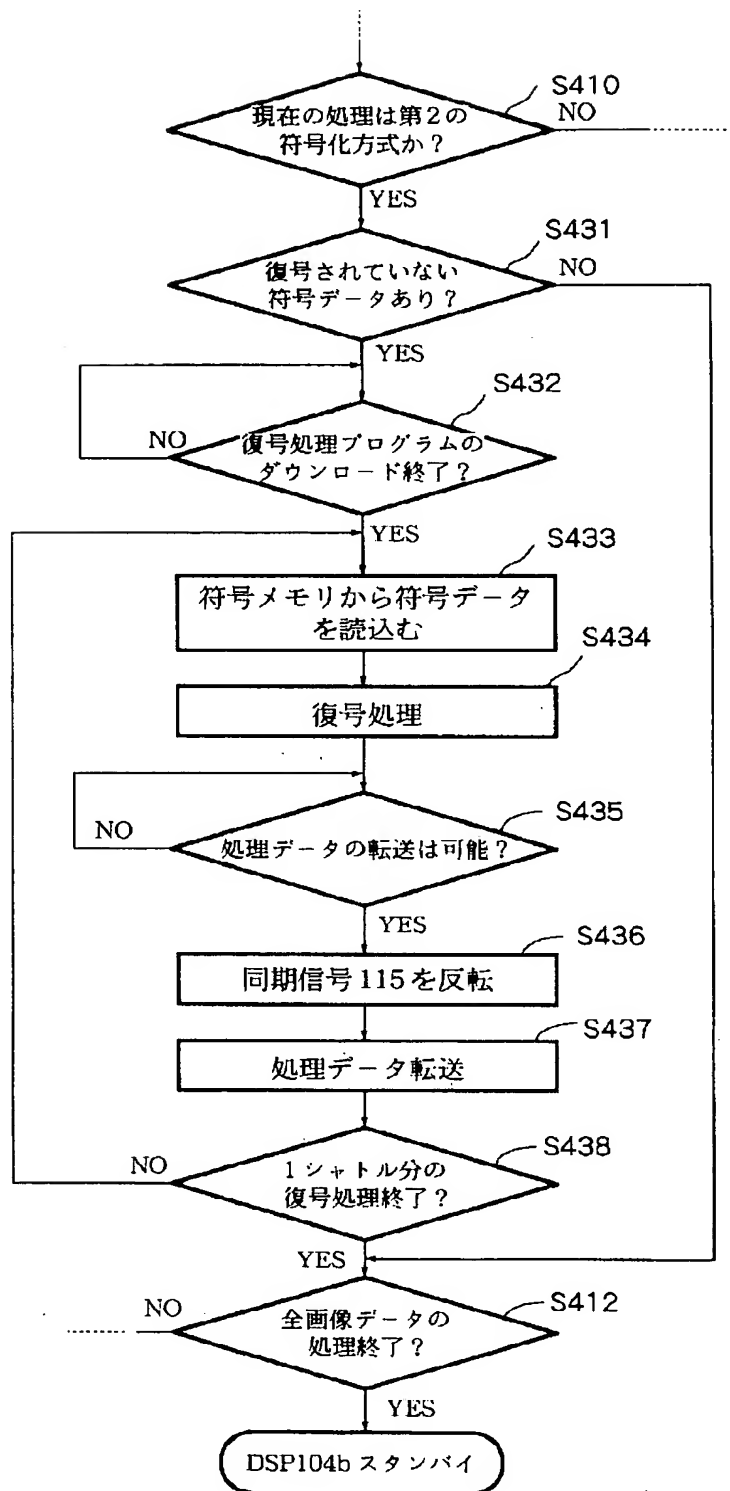
【図 21】



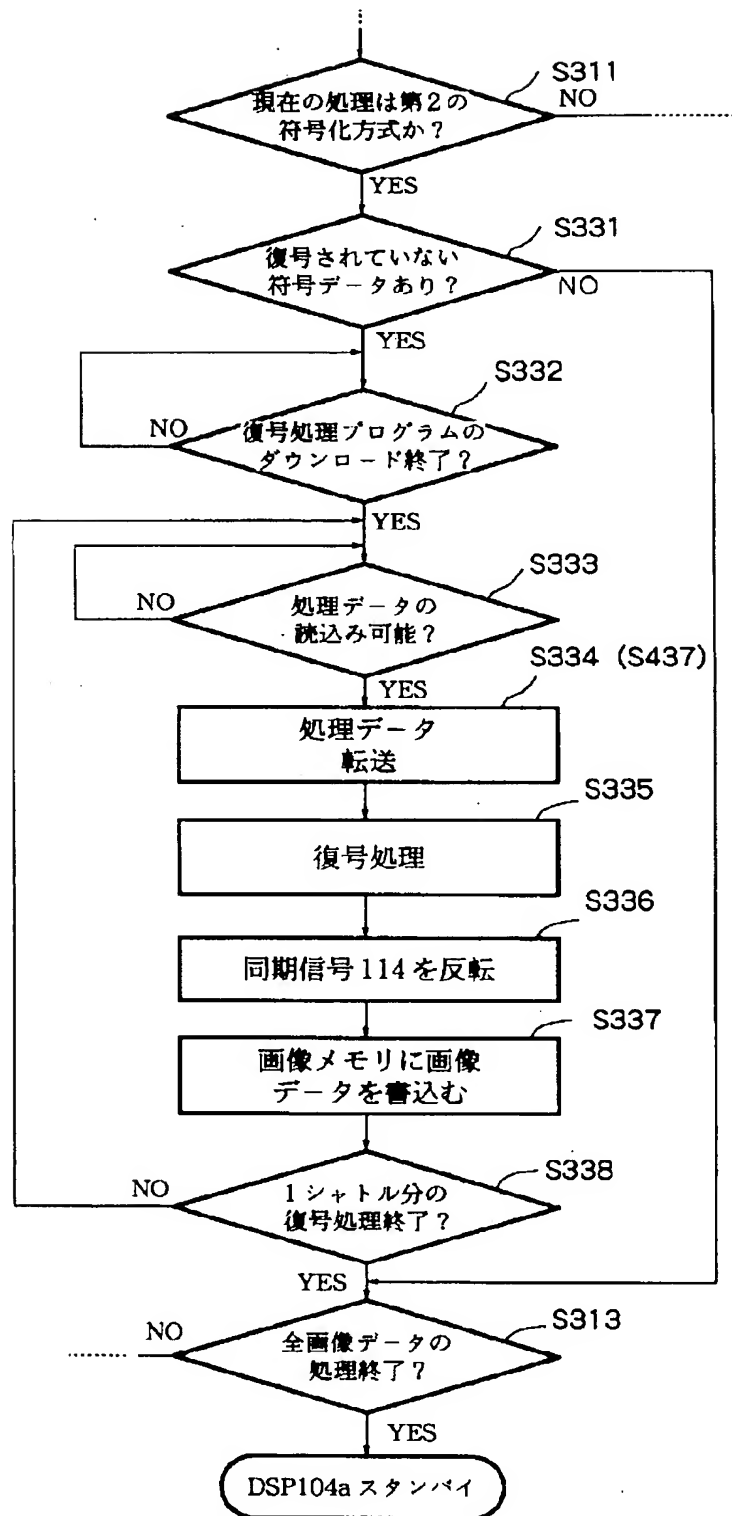
【図 22】



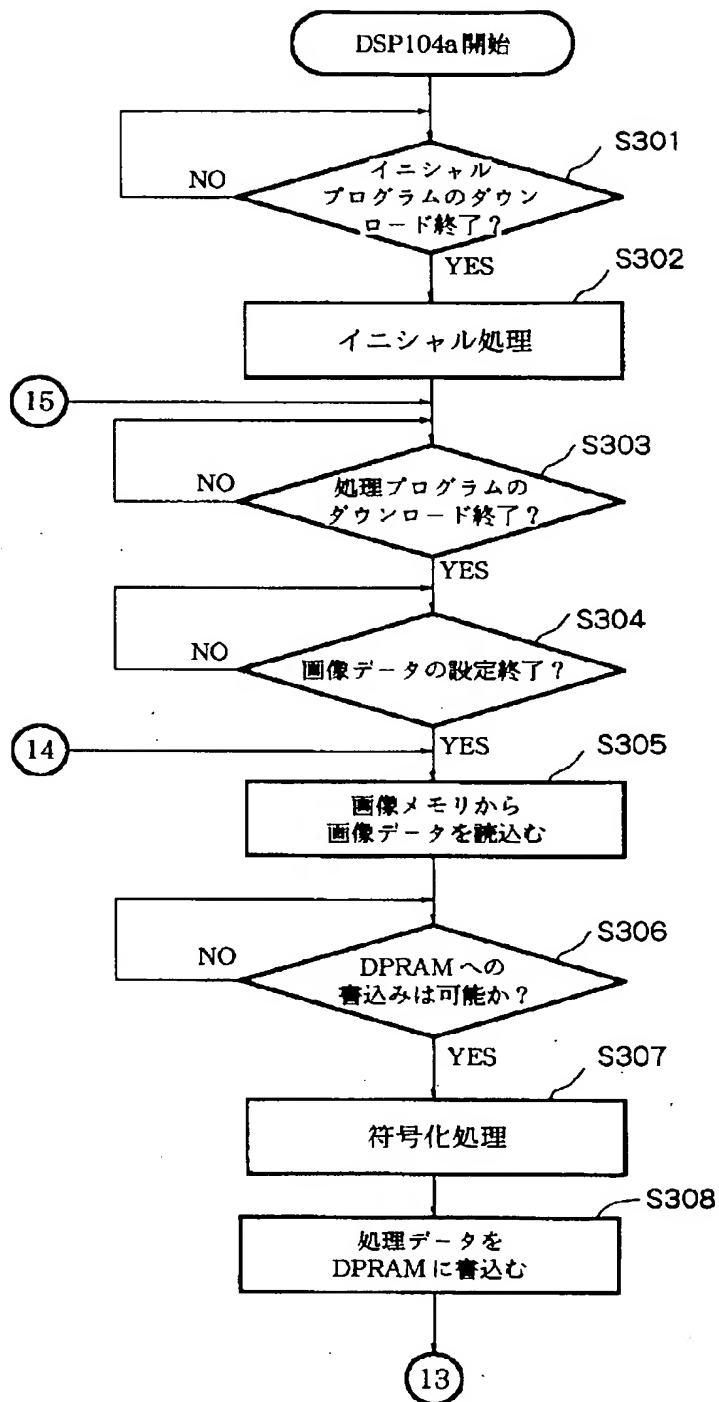
【図 23】



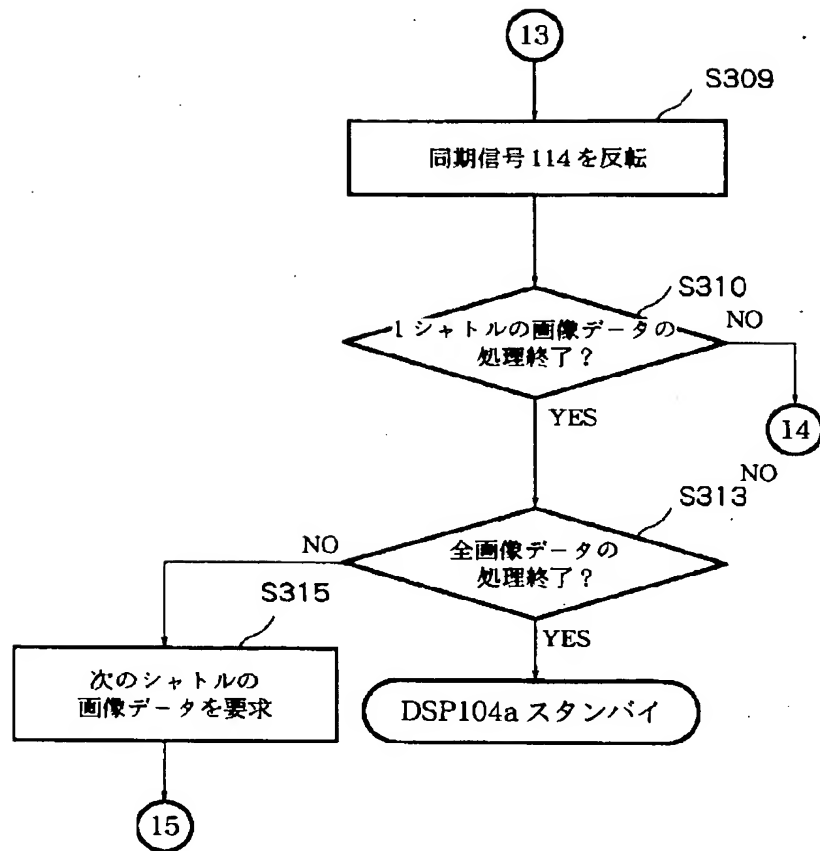
【図 24】



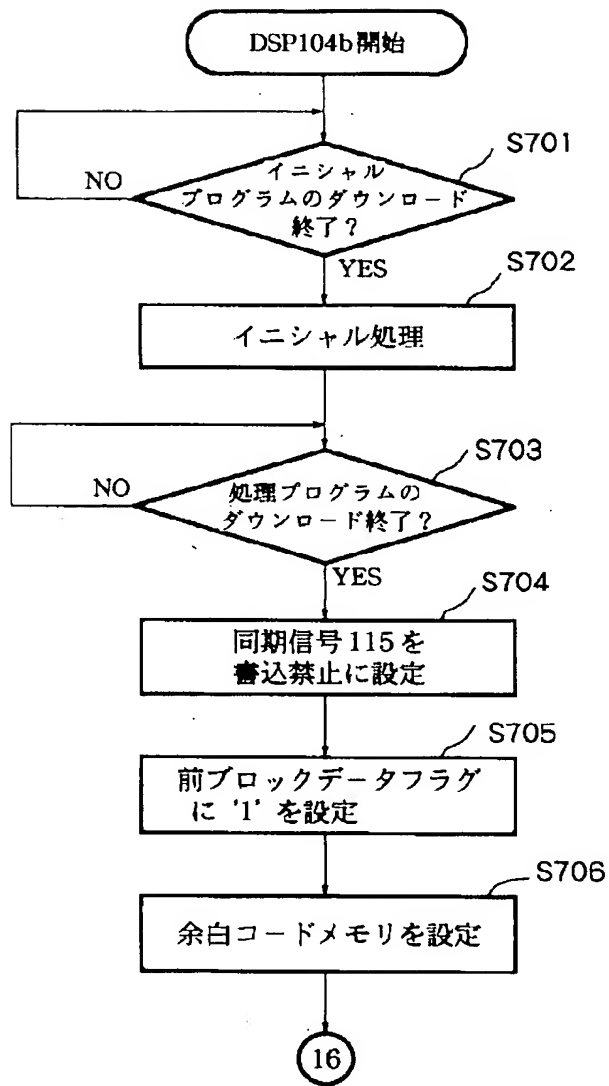
【图 2 6】



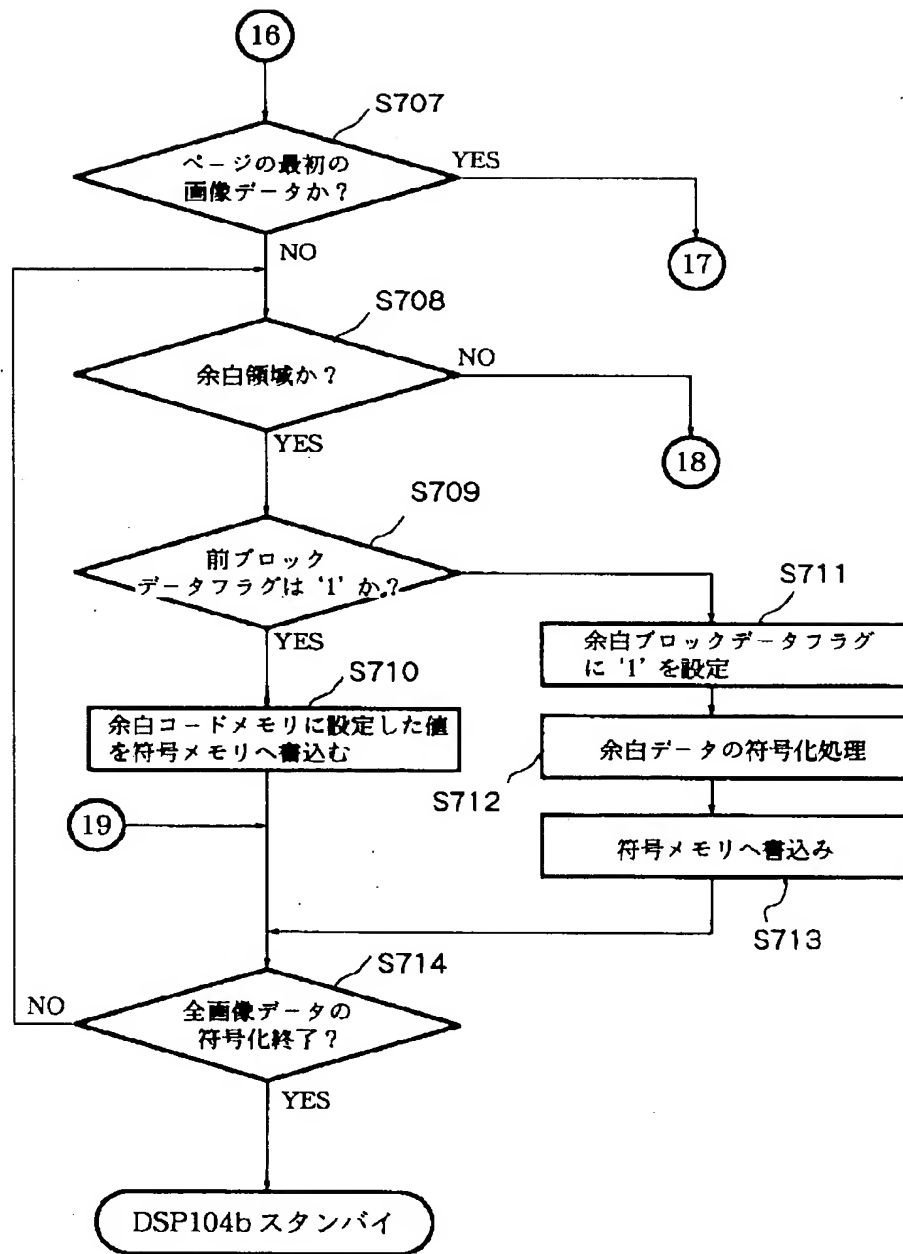
【図 27】



【図28】



【図 29】



【図 30】

